Detections and Applications of Saliency on 3D Surfaces by Using Retinex Theory

by

Yitian Zhao

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Department of Computer Science

ABERYSTWYTH UNIVERSITY

September 2013

"What you see is what you get."

Lord Alan Sugar

Abstract

Unlike traditional 2D images, which are projections of the real world onto a two-dimensional surface, 3D images express the geometry of the objects of interest directly in terms of a set of points, a mesh, or a surface composed of points with three-dimensional coordinates. The size or shape of this 3D information of an object may be computed almost directly from its three-dimensional representation. 3D imaging geometry essentially simulates human binocular vision, and enables a direction acquisition of the depth information from the camera to the object of interest. It finds a variety of applications ranging from reverse engineering, urban planning and simulation to computer games.

With the evolution in recent years of more modern technologies and devices, there has been enormous growth in the number of 3D models/3D images and their availability to various communities. Examples include the National Design Repository, which stores 3D computer-aided design (CAD) models for tens of thousands of mechanical parts; and the Princeton Shape Benchmark (PSB) with 36,000 everyday objects represented as polygonal surface models.

Most of the latest scanners can generate a huge number of data points within a limited time (a matter of minutes). Even a single scan might contain millions of points, which often leads to expensive computation and storage. The development of relevant software has not matched that of 3D hardware. As the complexity of these data points has increased, the digital representation of the real world objects has become more accurate, but there is a trade-off between degree of accuracy and the cost of processing and storage of these models. Therefore, reduction of information content or simplification of the 3D data points is useful for efficient processing, and necessary for visualization in some cases. In the course of a thorough review of the relevant literature, we found that the existing simplification algorithms perform inadequately, especially at a very high simplification rate. In recent years, the notion of human visual perception has been explored with a view to aiding simplification. With a view to retaining the important surface features and details, the selection of samples is now guided both by geometric properties and by the visual attention properties of the surface. Thus, as the criteria of the simplification or interest points detection, salient regions and non-salient regions can be processed separately, preserving more vertices or facets from salient regions, while selecting fewer vertices or facets from non-salient regions (in our proposed interest points ddetection method, only select points from salient regions). The estimation of the perceptual properties/saliency of the target object is thus a very important preprocess for simplifying highly complicated 3D models.

In this dissertation, a surface smoothing and two novel saliency detection methods on 3D models are proposed. The acquired data usually contains imaging noise, due to low reflection or specular reflection, occlusion and depth discontinuity. Sometimes a

rough surface is generated due to the rapid changes of orientation and vertex locations of reconstructed surfaces caused by noise introduced in the process of surface scanning, image registration and integration. Hence, an extended non-local means filter has been proposed in the case of a 3D surface. To the best of our knowledge, there is no previous work on non-local means filtering of mesh with B-spline optimization. As we know, the non-local means filter takes advantages of the high degree of redundancy of any natural image. For a given pixel, the restored gray value is obtained by the weighted average of the gray values of all pixels in the image; each weight is proportional to the similarity between the local neighborhood of the pixel being processed and the neighborhood corresponding to the other image pixels. With this filter, a smoother version can be robustly obtained, since it defines the similarity between patches of pixels, rather than between the individual pixels themselves. However, when extending the 2D non-local means filter to the processing of a 3D mesh, a problem arises in the determination of the similarity neighborhood. 2D images usually have a regular structure, which in most cases it is not true for a mesh due to variations of sampling density in the range scanning process. In this work, the B-spline is employed to determine the similarity neighborhood, which in turn generates the control net for the input mesh. The advantage of using B-spline surfaces is that the underlying control net is topologically similar to the image grid structure.

The first saliency detection approach adapts *Retinex* from a 2D image enhancement technique to analysis of geometry or shape variation in 3D models. *Retinex* investigates the theory behind the constancy of color. It explains from a psychological perspective why the colors perceived by human beings are relatively stable, usually irrespective of illumination conditions. *Retinex* has also been imported into the computer vision field, in which the captured data are often unsatisfactory due to low contrast - either locally or globally - caused by too weak or too strong illumination, or even shadow. *Retinex* is extended here to enhance 3D shape information and aid analysis of global shape and local geometrical details. Normally, human perception and objective information with respect to vision are not in agreement. The human brain interprets an image of a 3D shape differently from how photo-sensors or scanners may sense it, by consciously correcting brightness and removing noise, shadows, glare, or reflections. After the application of Retinex, the 3D shape, component or surface may be represented more faithfully to the original, simulating the effect of human visual systems. After using the *Retinex* to enhance the surface, a random center-surround saliency detection is proposed. The main structure of our saliency system is based on the general layout of psychological attention models, and it improves and extends the concept of *mesh saliency*, integrated for more accurate detection of importance/saliency of points.

While the first saliency detection approach is powerful for the characterization of the importance/saliency of points, it may be affected by imaging noise or depth discontinunity, leading to the salient regions being only partially detected. To overcome this shortcoming, a second method is proposed that measures similarity based on patches, rather than individual points. This saliency detection approach is an extension from the first saliency detection method. Based on observations from studies of biological vision, we know that the human vision system is sensitive to contrast in visual signal. It is widely believed that human cortical cells may be hard-wired to respond preferentially to high contrast stimulus in their receptive fields. Therefore, if a specific contrast for the 3D surface is generated, it may also be used to illustrate the difference in the geometry or topology that makes the local details or global shape distinctive. In this study, by combining Retinex-based Importance Feature, and Relative Distance, a weighted dissimilarity map is obtained to generate the 'surface contrast'. The dissimilarity map is estimated as the sum of difference between geometric invariance of different points inside two patches, inversely proportional to their Euclidean distance. Subsequently, the global nature of salient regions are captured by considering the symmetric surround saliency. As noted above, as we know humans pay more attention to those image regions that contrast strongly with their neighbors. To determine the region-based saliency, a region-growing segmentation is employed to segment the surface. The results show that the proposed approach has the ability to locate the distinctive regions faithfully.

In order to validate the proposed saliency detection methods, the detected salient regions have been applied to simplification, and interest points detection. A large number of experiments based on real data captured by Minolta Vivid 700 range camera show that more details have been retained in the process of surface simplification, the detected interest points are more repeatable - useful for the representation of the geometry and detail of the object of interest. In addition, the comparative studies also show that the propose techniques outperform the state-of-the-art methods and have clear advantages.

Acknowledgements

Completing this thesis is a multi-stage process that involving taking classes, studying previous research, and exploring new areas of science. Numerous people provided guidance and encouragement along the way, from the beginning through my time at Aberystwyth University.

I would like to express sincere appreciation to my supervisor Dr. Yonghuai Liu, for his excellent mentorship. He contributed in various ways for completion of this project. His broad knowledge and enthusiasm for exploring new research fields have had a great influence on me.

I want to thank my second supervisor, Dr. Frederic Labrosse. I appreciated his advices and help he gave me during study toward my Ph.D. degree. Recently, he provided feedback on my research and reviewed this dissertation. I am grateful to Prof. Reyer Zwiggelaar, head of the VGV group, for proofreading some of my publication papers, for providing travel grants to me to attend several academic conferences, and thus I can collect the latest research information and embed into my own work. Many thanks also to the other staff in the Department of Computer Science, for their friendliness and innumerable help.

I would also like to thank all the members of C57 and F01, especially Drs. Zhili Chen, Ziming Zeng, Ran Song and Bashar Al-Rjoub for helping me meet the target of this project.

I would like to thank my housemate, for her support: Emma Towner and her family. They were always willing to help and gave their best suggestions. Moreover, to express gratitude and best wishes to my great friends: Miaomiao, Min Jie, Long Jie, Xiangxiang, Guaiguai, Min Ge, Xiao Ma's, Yuanyi, Xiao Su Gege, Yongfeng, Teacher Song's, Boss Jin, Lao Diao, Zhuoke, Juan Jie, Lin Meimei, Xiao Shimei, Pan Lilan, Mr. Lou, Gui Ge, Teacher Chen, Zheng Shimei, and whoever had provided any factor of fun in my PhD life.

Finally, my heart overflowed with gratitude to my family. The support of my family is invaluable. My parents Sheng Zhao, Youguang Zhao and brother Dr. Yifan Zhao, sister-in-law Dr. Huimin Guo were always there cheering me up and stood by me through the good time and bad time. I would not have met this stage without their unconditional love and constant support.

Contents

List of Figures

List of Tables

Dedicated to my parents, Sheng Zhao and Youguang Zhao

Chapter 1

Introduction

Three-dimensional (3D) technology was once exclusively the domain of skilled computer-graphics developers and gamers with access to high-end software. A complex mixture of colors, textures, virtual lighting and perspective are necessary to make images appear three-dimensional. In the past, the 3D technology was expensive and difficult to use. Today, this scenario has changed. 3D technology is becoming more and more popular in everyday life. Many 3D products have been developed, such as 3D printer, 3D camera, 3D TV, 3D games, 3D films and so on.

All the technologies mentioned above are centrally concerned with 3D imaging. 3D imaging aims to capture the geometry of objects in terms of 3D coordinates, and in consequence can simulate human vision in the real world within a computer. The computed set of data points, with or without accompanying colors or textures, is referred to by various names, such as '3D model', '3D scan', or '3D image'. Compared with a traditional 2D image, the shape and size of a 3D image may be straightforwardly computed from its 3D coordinates. The output of a 3D imaging process may be analyzed and processed to extract information that may then be used in support of a wide range of applications, such as object recognition, robot navigation, urban reconstruction, and clinical procedures in medicine.

1.1 Applications of 3D imaging

As of today, a wide range of applications have been developed. This section presents a brief overview of the most important fields in which 3D imaging may be fruitfully used.

1.1.1 Industrial applications

3D imaging techniques deal with typical measurement issues in the industrial field: machined surfaces quality control; dimensional measurement; and the reverse engineering of complex shapes.

- **Machined surfaces quality control** [5, 6, 7]. 3D imaging techniques are the best suited to carry out the measurement processes in the surface control field [8]. Typical applications are the measurement of roughness and of waviness, and the gauging of the 3D topology of machined surfaces, as 3D imaging has the ability to represent a surface in measurement intervals from a few microns to tens of millimetres, with resolution from a few tens of nanometres to tens of microns. The top image of Figure 1.1 shows a 3D scanner at work in quality control of a machine element. Examples of market-available surface quality control products include the LNS 2.3/4.4 Dynavision (LMI Inc.) and LTS 15/3 Daynavision (http://www.dosmar.fi/doc/automation/LDSApps.pdf). These patented sensors measure resolutions as fine as 5 nanometers, making them ideal for quantifying the surface roughness of any highly reflective material.
- **Dimensional measurement** [9, 10, 11]. As the dimensional measurement of complex surfaces involves the 3D acquisition of shapes in the range from millimetres to meters, with measurement resolution from one hundred microns to a few centimetres, this is a very wide-ranging field of application. Examples of firms making commercially available dimensional measurement products include: the Ortigo 200 system (Cognitens Inc) (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3280764/pdf/sensors-09-00568.pdf.); and the R Scan (Hexagon Inc) (https://store.hexagonmetrology.us/). In these two systems, a scan arm is equipped with laser stripes to fully control the rotation and the translation of the optical head. The sequence of image acquisition is determined on a case-by-case basis by the dimensions and shape of the object to be imaged [8].
- **Reverse engineering of free-form shapes** [12, 13]. The technique of using sensor data to create a set of specifications and drawings for a geometric model directly from the inspection of an object is called reverse engineering. The current industrial technology for precise measurement of 3-D objects involves contact scanning by Coordinate Measuring Machines (CMMs) [14]. A survey of the techniques developed in this field is in [15]. This technique is precise, and widely used for the creation of surface models.

The bottom images in Figure 1.1 show the performances of 3D techniques for the quality control of machine elements and for the reverse engineering of free-form shapes.

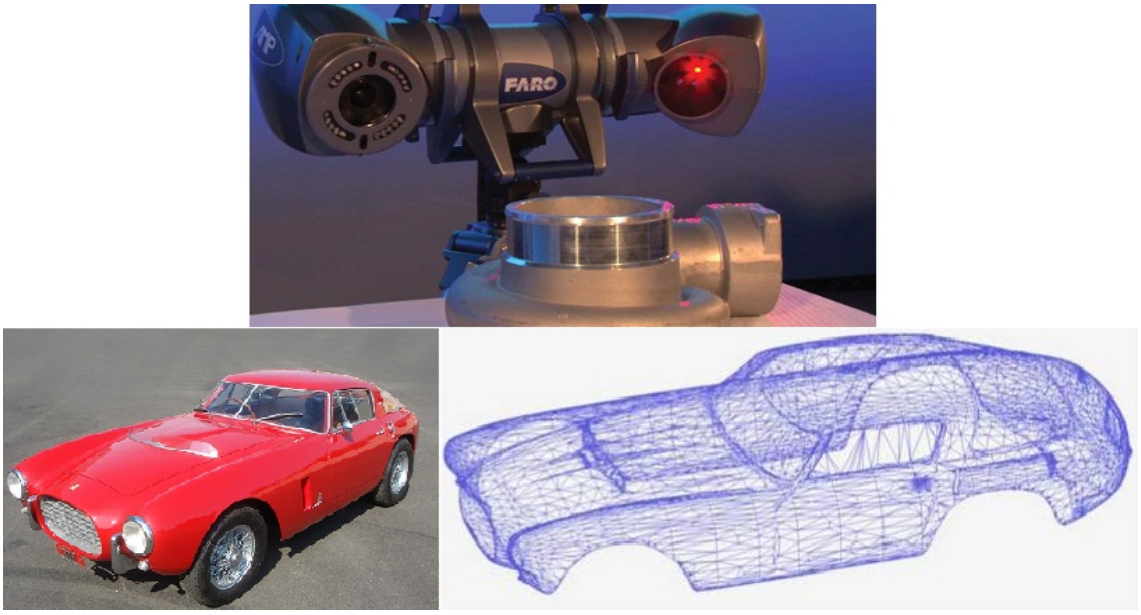


FIGURE 1.1: Example of 3D imaging application in the industrial field. Top: scanning a machine element. Bottom left: the historic Ferrari 250MM racing car. Bottom right: The racing car as represented by a triangulated 3D mesh [8].

FIGURE 1.2: Example of the use of a 3D imaging application in the cultural heritage field. Left: the original sculpture of *david*. Right: reconstructed 3D model.

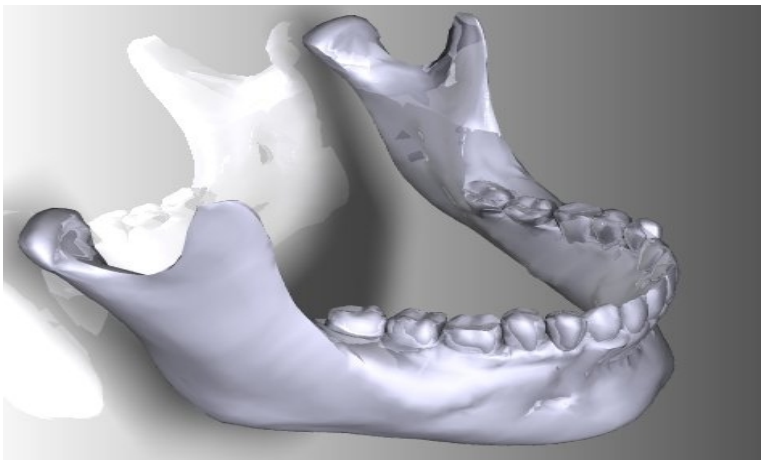


FIGURE 1.3: Example of 3D a imaging application in the medical field: a reconstructed 3D model of the teeth of the lower jaw. (http://www.the3dstudio.com/product_details.aspx?id_product=522913)

1.1.2 Cultural heritage applications

In the last few years, the use of 3D imaging products for the contactless acquisition of culture heritage data has attracted increased attention from museums and archaeological specialists. They have similar objectives: data acquisition, modelling and visualization are required within the cultural frame, to monitor new archaeological finds [8]. Moreover, reconstructed 3D models help specialists to make physical copies, repair damaged regions, and even to simulate the probable appearance of incomplete parts of the historical relics. Figure 1.2 gives an example of a reconstructed model produced from a 3D scan.

1.1.3 Medical applications

Medical applications that benefit from 3D scanning include, but are not limited to:

- Dental: making braces, retainers, and mouth guards.
- Facial: making form-fitted face masks for treating burn victims.
- Hands: making customized gloves for patients.
- Other body parts: making prosthetics; for example, for the leg and spine.

As the 3D scanner can capture a single scan in just a few seconds, the 3D measurements can be obtained quickly, and the patient's condition may be diagnosed much more rapidly.

1.2 Techniques for 3D Imaging

There is a long history of 3D imaging that can be traced back to ancient times. Humans have tried to capture their 3D environment and important aspects of social life in wall paintings, some of which are thought to date back 32,000 years. In the modern period, a stereoscope invented by David Brewster in 1844 could take 3D photographic images. In the succeeding years, there were further improvements in the technology: Louis Jules Duboscq improved on the stereoscope, and a famous 3D picture of Queen Victoria was displayed at the Great Exhibition in 1851. A stereo animation camera was invented in 1915: and the first publicly-shown 3D movie was made. In the 1960s, a new technology known as Space-Vision was released. During the last fifty years, rapid development in the fields of solid state electronics, photonics, computer vision and computer graphics has meant that the capturing and recording of detailed shapes by optical techniques. Now, large amounts of 3D data can be captured by reliable, accurate, high-resolution 3D imaging systems.

Typically, 3D imaging sensors operate by projecting (in the active form) or acquiring (in the passive form) electromagnetic energy onto or from an object, followed by recording of this transmitted or reflected energy. In this section, we briefly introduce modern 3D imaging technologies by grouping them into two categories: passive and active 3D imaging, respectively. A survey of 3D imaging techniques is also given in [16].

1.2.1 Passive 3D Imaging

Passive 3D imaging must be taken from standard 2D images, and relies on images of the ambient-lit scene alone, without the help of the artificial lighting. There are a set of techniques which develop this approach to imaging based on different visual cues: shape from focus, shape from shadows, shape from texture, and shape from stereo disparity.

- **Shape from focus** [17]: by exploiting the depth of focus phenomenon, one of a pair of camera lenses can be used as a range finder. The target image is blurred by an amount proportional to the distance between points on the object and the in-focus object plane.
- **Shape from shadows** [18]: uses the shades in a greyscale image to infer the shape of the surfaces, based on the reflectance map: the image intensity with changing surface orientation may then be linked by this map.

- **Shape from texture** [19]: assumes that the object is covered by a regular surface pattern; then attempts to find possible transformations of texture elements in order to reproduce the object surface orientation.

- **Shape from stereo disparity** [20, 21]: this is the most frequently-adopted technique. (Please refer to [22] for a survey of stereo methods.) 3D data acquisition and object reconstruction can be performed using stereo image pairs. Stereo, also known as stereovision, was developed in the 1970s, using a special anamorphic lens that widened the picture using a series of polaroid filters. Stereo imaging involves using two cameras to generate depth information about a scene through a comparison of corresponding features in both stereo images. Depth information can be obtained by examining the relative positions of objects in each of the two perspectives. However, stereo imaging is sensitive to noise and textural appearance, since it is an intensity-based method and vulnerable to imperfections in the object's surface. A disparity (difference) map can be formed from the combination of the two images that will allow a system to make decisions based on the distance of objects from the camera.

This kind of approach is based on knowledge of the scene structure and the internal and external parameters of the cameras used. Such a system is illustrated in Figure 1.4(a).

Furthermore, in 1981 Long-Higgins improved the stereo system by using Essential Matrix, in which eight or more image correspondences are given for a stereo pair; and in 1992, Faugeras et al. [23] extended the use of Essential Matrix [24] to uncalibrated cameras through Fundamental Matrix, which requires no camera-intrinsic parameters to be known in advance.

The advantages of passive imaging include: inexpensive hardware (two cameras); ease of accommodating motion; and the fact that it works intuitively, by analogy with human vision. There are however four main difficult challenges in the use of passive 3D imaging: occlusions (a point on the scene viewed by one camera is not necessarily viewed by the other); high computational complexity; inability to distinguish single regions of homogeneity - i.e., false detection of internal difference, leading to identification of multiple regions; and occasions on which the same object feature generates different values in each of the stereo images.

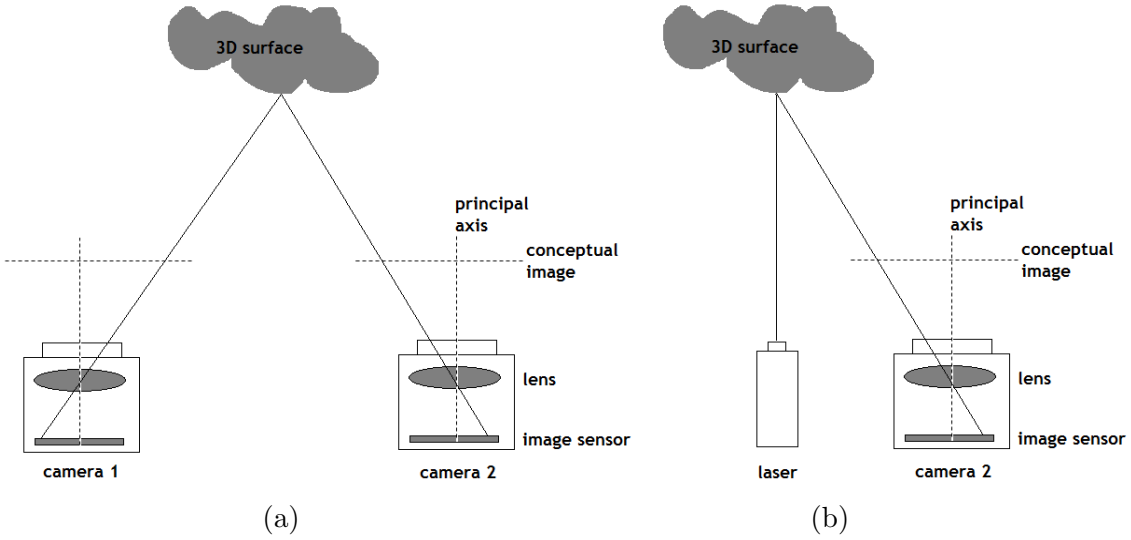


FIGURE 1.4: Two different 3D imaging types. (a) Passive 3D imaging. (b) Active 3D imaging.

1.2.2 Active 3D Imaging

Active 3D imaging avoids some of the difficulties of passive techniques by introducing controlled additional information: artificial illumination, usually either a spatially coherent light source (e.g. laser) or an incoherent one (e.g. halogen lamp). It is possible to generate a model of a surface geometry even if the surface appears featureless to human eyes or unassisted camera by using an artificial light source. In the mid-1990s, the 3D laser range scanner had been developed and had overcome many stereo imaging issues, yielding more accurate solutions to the problem of range image acquisition. The most significant characteristic is that the system consists of a single camera with an active light source - a laser. The output of laser range scanners is usually a set of structured data points with or without reflectance strength information, depicting the reflectance characteristics of the 3D objects of interest. (Excellent reviews of optical methods and range sensors for 3D measurement are presented in [25].)

Figure 1.4(b) details the technique of active 3D imaging. The captured 3D data can be made relatively revealing of ambient illumination and surface color. This technology enables easy acquisition of both geometrical and/or photometric information about the objects of interest in the form of range and/or intensity images. Laser scanning technology offers the fastest and most automated way to collect height or distance information.

Active 3D imaging systems may be based on any of three different measurement principles: laser triangulator, time-of-flight, or interferometry.

- **Laser triangulator** [26, 27], uses a laser source to generate a narrow beam to the target. The back-scattered beam is then imaged at the camera sensor. By means of simple trigonometry, the position of the target is calculated. The measurement of the surface is achieved by scanning. A more optical triangulation principle has also been exploited, such as the use of laser stripes. The most significant advantages of laser triangulators are their accuracy, and their relative insensitivity to illumination conditions and surface texture effects.

- **Interferometry** [28]: based on the superposition of two beams of light. It operates by projecting a spatially- or temporally- varying periodic pattern onto a surface, followed by mixing the reflected light with a reference pattern. The measurement resolution is very high, since it is a fraction of the wavelength of the laser radiation. In consequence, surface quality control and microprofilometry are the most explored applications.

- **Time-of-Flight** [29, 30, 31]: which can be traced back to the era of RADAR, which is based on comparison between a transmitted radio wave and its reflection. The underlying idea is that a receiver detects a reflected pulse, and suitable electronics measures the roundtrip travel time of the returning signal and its intensity. With the development of lasers in the late 1950s, it became possible to image a surface with angular and range resolutions much higher than is possible with radio waves.

The advantages of active 3D imaging include: the generation of dense data; and the production of much more robust and accurate data than is produced by passive techniques. The main drawbacks of such 3D imaging are: the requirement for the introduction of artificial light into the scene; relatively high cost; and the fact that none of these techniques is analogous to human vision.

1.3 Representation of 3D Data

The representation of the 3D data is the essential step for a number of important applications, such as computer-aided geometric design, visualization and graphics. There are several different approaches to 3D representation, including raw data, surface and solid. In this section, we limit ourselves to introducing the first two types of 3D data representation.

1.3.1 Raw Data

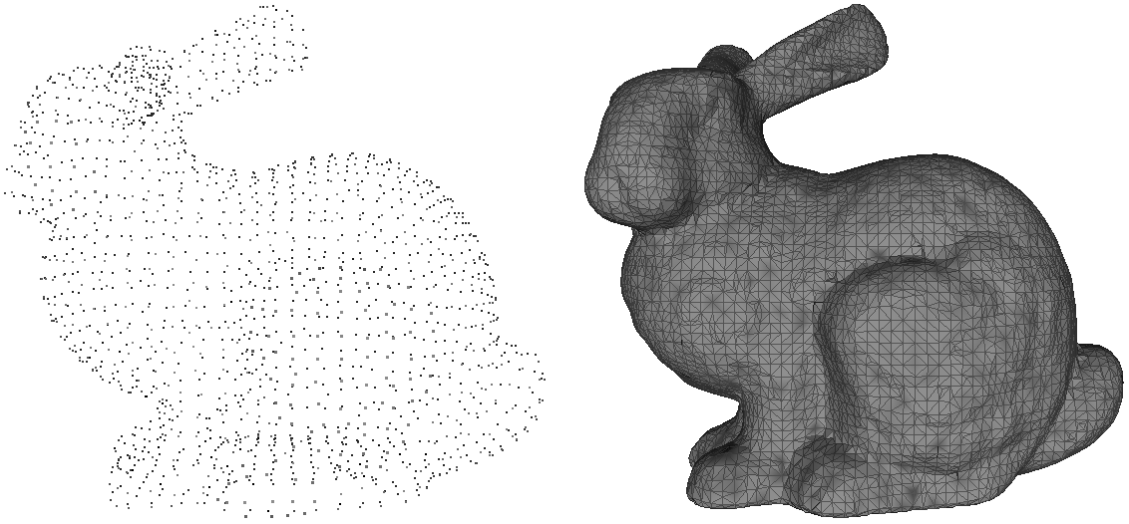
The first type is raw data (i.e. as delivered directly by a 3D sensing device). The raw output of a 3D sensor can take any of a number of forms, such as points, depth maps, and polygons.

- **Point cloud** representation is the simplest form of raw data [32]. Typically, a point cloud of n points is stored as an $n \times 3$ array of floating point numbers. They are generated by laser range scanning devices. Sometimes a more constrained representation may be used when point cloud vertices adhere to an underlying structure: namely a grid with arbitrary sampling, in which the vertices are stored as an $m \times n \times 3$ array, for each point $i = 1...m$, $j = 1...n$ and corresponding 3D vertex $[x(i,j), y(i,j), z(i,j)]$. An example of visualization by means of a of point cloud may be seen in Figure 1.5(a).
- **Depth map and range image** [22] are defined as a structured point cloud and orthogonal projection of the 3D vertices over a 2D image plane, if the sampling of points in the $x - y$ plane is viewer-centered. They can be visualized as grayscale images, in which image intensity indicates the distance to the surface.
- **Polygon soup** is a set of unstructured polygons, each of which connect vertices together, but which are not themselves connected in a coherent structure [33, 34].

1.3.2 Surfaces

Another type of 3D data representation is surface. The vast majority of 3D data representations are surface representations. The most common surface representation is the polygon mesh: this is comprised of a set of 3D vertices, connected together to form triangular or polygonal faces, which in turn represent or approximate a 2D manifold embedded in 3D space. An important distinction is whether the mesh is closed or open.

- **Trianglar mesh** is the most popular mesh, and there are several alternative triangulation approaches, such as step discontinuity constrained triangulation [35], statistical discontinuity constrained triangulation [36], angle discontinuity constrained triangulation [37], and Delaunay constrained triangulation [38]. Figure 1.5(b) shows a reconstructed triangulated mesh of model *bunny*.
- **Quadrilateral mesh** is another polygon mesh type: it can be easily converted to a triangular mesh by diagonally subdividing each quadrilateral.



(a) (b)

FIGURE 1.5: Two different 3D data representation types. (a) Point cloud. (b) Surface.

- **Subdivision surfaces** are used to represent a smooth surface using a low resolution base mesh and a subdivision rule.
- **Implicit surface** [39] is the set of all points $[xyz]^T$ that satisfy the function $f(x, y, z) = 0$. For applications involving visualization of physical effects such as fluid dynamics, implicit surface is a natural representation.

1.4 Motivation

With the continuing evolution of modern technological devices, in recent years, there has been enormous growth in the number of 3D models; more and more complex surface models have been generated. As the complexity of these models has increased, their visual approximation to the real-world objects has improved, but there has been a trade-off between the cost of processing and storing these models and superior visual approximation: most of the latest scanners generate huge numbers of data points within a limited time. Even a single scan may contain millions of points, which leads to expensive computation and storage.

Figure 1.6 shows examples of realistic objects whose representation as 3D images generates complex models with huge number of polygons. Other examples include those in the National Design Repository, which stores 3D computer-aided design (CAD) models for tens of thousands of mechanical parts; the Stanford 3D Scanning Repository, which stores several well-known 3D models that have been used by academic researchers,

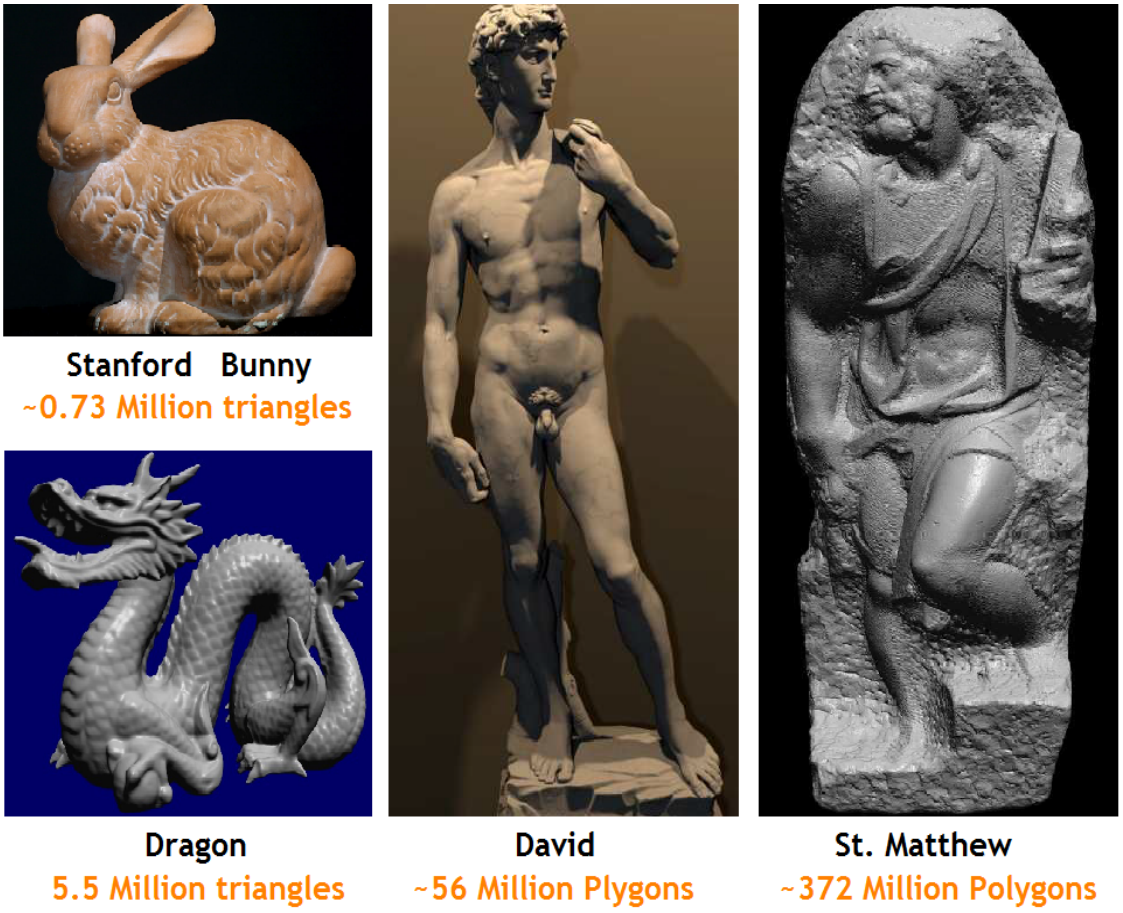


FIGURE 1.6: Examples of 3D models of realistic objects with huge numbers of polygons. (http://graphics.stanford.edu/data/3Dscanrep/)

the Protein Data Bank (PDB), with atomic positions for tens of thousands of protein molecules; medical collections, such as the Visible Human; and the Princeton Shape Benchmark (PSB), with 36,000 everyday objects represented as polygonal surface models [40].

However, the development of relevant software has not matched that of 3D hardware techniques, and we are now facing great challenges in storage, editing and transmission. Most of the 3D models in the aforementioned datasets are large: for instance, *Happy Buddha* in the Stanford 3D Scanning Repository contains 4,586,124 points, generating about 9,200,000 triangles, which leads to an 89 MB storage requirement. This is a heavy demand for a single 3D model. Therefore, it is essential to find ways of reducing the information content of our 3D models.

We define simplification as the process of reducing the number of faces and vertices of a given input 3D model. There have been many attempts at producing a viable simplification technique over the past few decades. According to the method by which

(a) (b) (c)

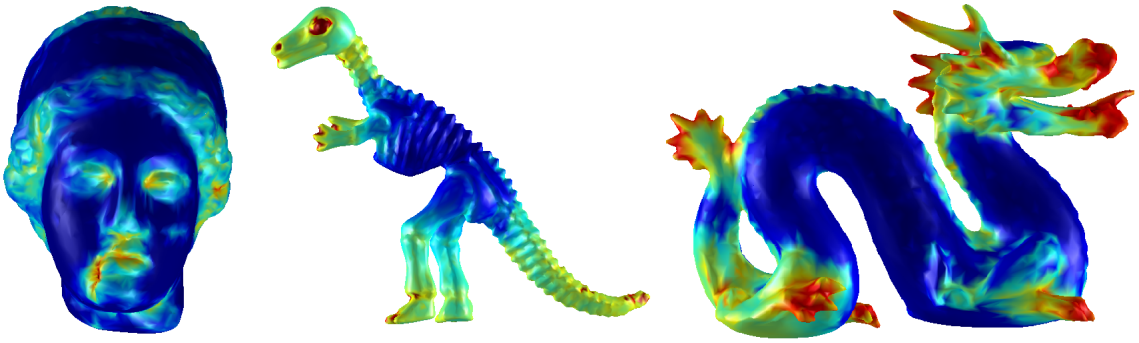
FIGURE 1.7: Model *Armadillo* from the Stanford 3D Scanning Repository, and under different levels of simplification [1]. (a) Original. (b) 20% of vertices. (c) 5% of vertices.

the vertex is manipulated, these techniques may be grouped into three main categories: vertex decimation [41], vertex clustering [42] and vertex collapse [43]. Vertex decimation aims to reduce the number of polygons. It removes vertices, edges and faces, thus creating a coarser mesh. The vertex clustering process determines the closeness of the vertices in the object space, and for those vertices found to be close to one another, a new representative vertex is created to replace them. Vertex collapse is an optimization algorithm. It iteratively selects the edge with the least error for collapsing, removes all adjacent vertices and faces, and re-triangulates the modified models. Regarding the importance of simplification algorithms, we make reference to a number of survey papers [1, 44, 45, 46].

These techniques have in common that after simplification, the total number of the vertices is reduced, while the global information content of the model is conserved. Most simplification algorithms distribute samples on the surface according to the geometric properties of the surface. Figure 1.7 shows such geometric-based simplification results, obtained by using the approach outlined in [1].

Most of the existing algorithms for mesh simplification perform poorly at very low levels of detail, as shown in Figure 1.7. In order to retain the important surface features and details, in recent years simplification guided by human visual perception has been explored [3, 47, 48, 49]. In this approach, the distribution of samples is guided both by the geometric properties of the surface and by the visual attention (saliency) properties of the surface.

Figure 1.8 shows detected visual attention (saliency) on a variety of 3D surfaces generated by the approach outlined in [50]: the non-blue areas indicate the salient regions.



(a) (b) (c)

FIGURE 1.8: Examples of salient region estimation on different 3D models [50]. (a) Venus. (b) Dinosaur. (c) Dragon. Red indicates high saliency and blue low saliency: other colors denote in-between values.

Thus, for the simplification or level-of-detail criterion, salient regions and non-salient regions can be processed separately: preserving more vertices or faces from salient regions and selecting fewer vertices or faces from non-salient regions. Therefore, the estimation of the perceptual properties of the target object might be a very useful pre-process for simplifying high-complexity 3D models.

1.5 Visual Saliency

> 'Everyone knows what attention is. It is the taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought... It implies withdrawal from some things in order to deal effectively with others.' (William James, Principles of Psychology, 1890)

What you see is determined by what you attend to. We are aware of only a small portion of the visual information that impinges on our visual apparatus, because the human visual system has limited capabilities, and cannot process everything that falls onto the retina. Instead, the brain relies on *attention* to focus on the most salient details. The mechanism in the brain that determines which part of the plethora of sensory data is currently of most interest - is *salient* - is called *selective attention* [51]. The cocktail party effect [52] is a well-known example in the field of selective attention. Due to the human need to deal with a high amount of sensory input from moment to moment, evolution has favoured the development of selective attention. A large amount of data is in general too large to be completely processed in detail and the range of possible actions at one and the same time is restricted: consequently, the brain has to prioritize.

The same problem is faced by computer vision systems: there are thousands, sometimes millions of pixel values to be dealt with in a single image, which leads to high computational complexity. The task becomes especially difficult if a system has to operate in real-time. An hypothesis had been proposed [53]: that computers or robots should automatically focus on the relevant data is even more important than for systems that are purely concerned with computer vision. As a result, investigation has begun into how the concepts of human selective attention might be exploited for computational systems.

1.5.1 The Human Visual System

We begin by providing a very rough overview of the human visual system (HVS). A human visual system model is used by image processing, video processing and computer vision experts in order to deal with biological and psychological processes. Thus, the HVS is referred to as the complex set of biological and psychological elements that allow human vision of the physical world. Up to now, the most well-known parts of our visual system have been the transducer - i.e., the retina - and the processing unit - i.e., the V1 cortex (also known as the striate cortex) [54, 55, 56].

When a light ray hits the eye, it will first pass through the cornea, then subsequently through the aqueous humour, the iris, the lens, and the vitreous humour before finally reaching the retina. Figure 1.9 shows a cross-section of the human eye and identifies its most important parts. In the retina, the light rays are detected and converted into electrical signals by photoreceptors. There are two categories of photoreceptor (`http://www.salk.edu/news/pressrelease_details.php?press_id=443`): cones and rods, named after their approximate shape. Cones are dedicated to photonic vision and colors. In dimly-lighted conditions there is an intermediate stage, called mesopic vision, in which both rods and cones are active. The rods are more abundant than cones: there are about 100 million in a human eye by comparison with about cones.

We are able to distinguish colors because there are three types of cones, distinguished by the spectral sensitivity curve: L (red-orange), M (green-yellow) and S (blue). Figure 1.10 shows the basic structure of the photoreceptors in the retina. It converts visual information into electrical signals and sends them to an intermediate layer.

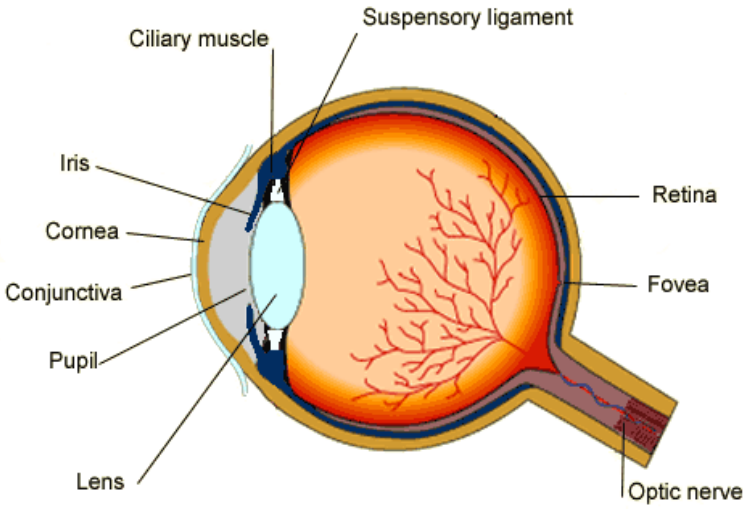


FIGURE 1.9: The structure of the human eye. (http://www.bbc.co.uk/schools/gcsebitesize/science/)

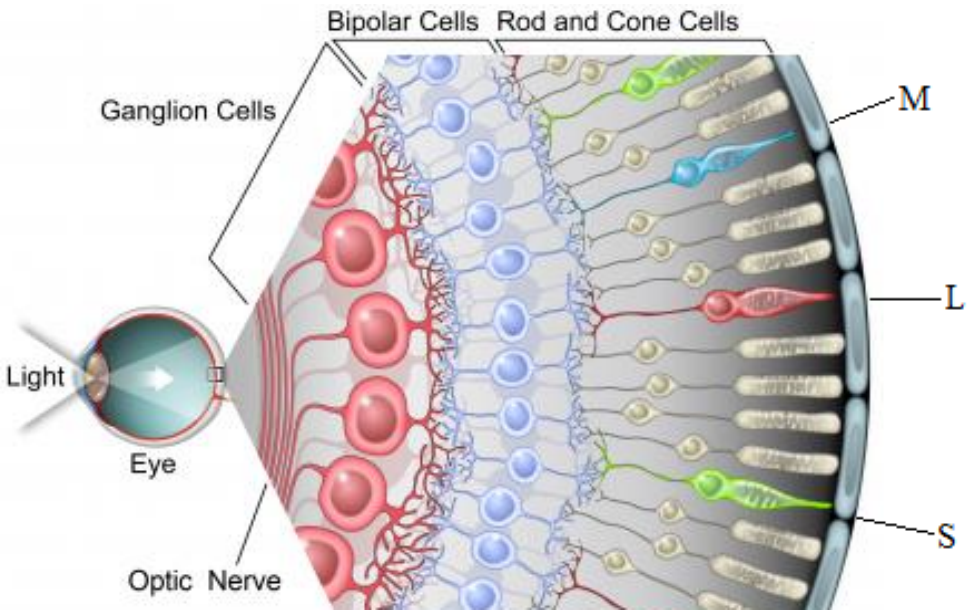


FIGURE 1.10: The basic structure of the photoreceptors in the retina. Photoreceptors in the retina convert visual information into electrical signals and send them to an intermediate layer, which in turn relays signals to the 20 or so distinct types of retinal ganglion cells. (http://www.salk.edu/news/pressrelease_details.php?press_id=443)

.

The retina may be considered as a preprocessing device that conditions the visual data for facilitated high level analysis. The primary visual cortex (V1) might be understood as a low-level visual information describer. Many simple feature computations take place during this operation. However, light acquisition and interpretation by the eyes/brain apparatus are very complex tasks that cannot be precisely modelled. Nonetheless, the joint analysis of visual phenomena from the biological and psychological points of view has recently produced several mathematical models that try to describe, with increasing level of accuracy, the behaviour of the HVS.

1.5.2 Computational Visual Attention (Saliency)

Visual saliency is a predictor of object regions which attract human attention: it indicates their relative importance and, being closely related to how we perceive and process visual stimuli, is under investigation by multiple disciplines, including cognitive psychology [57, 58], neurobiology [59, 60], and computer vision research [61, 62].

In the computer vision and robotics field, there is growing interest in the development of a selection mechanism that would determine the most relevant elements within a large quantity of visual data. Visual attention/saliency is the description or prediction of the observed region of interest by the human visual system, and it is one of the mechanisms underlying selective attention.

Many computational attention systems have been developed during the last 20 years. Most of these systems are considered to be based on psychological and neurobiological concepts and theories. In considering the psychological dimensions of human vision, most research addresses the understanding of how attentional mechanisms work, whether through psychophysics experiments in psychology or through neural recordings in neurophysiology. These types of models tend to focus on high-level justifications for specific attention mechanisms, and do not necessarily translate into computer vision algorithms [63].

From a psychological perspective, visual attention is categorized into two types: top-down attention (task-specific) [64] and bottom-up attention (stimulus-driven) [65]. The former type is goal-oriented and makes use of prior scene or context knowledge to identify the most salient regions. This approach demands a more complete understanding of the image context, resulting in high computational costs, as it is task-dependent. Top-down visual saliency involves the processes of feature learning and saliency computation. In psychophysics, top-down influences are often investigated by so-called cueing experiments, in which the cue directs the attention to the target. The cues have the ability

to indicate where or what the target will be. (For top-down approaches, the reader is referred to [66, 67].)

The bottom-up attention model is characterised by unique features, abrupt onset and appearance of new perceptual onset. It is usually attributed to early vision: i.e., to a model that responds to simple, and often local image features, such as a bright area in a dark scene [68]. (For examples of bottom-up approaches, we refer to [63, 65, 69, 70].) They are easy to implement, and require no prior knowledge of the scene. Figure 1.11 shows results of computational visual attention on 2D images.

Visual saliency is believed to drive human fixation behavior during free viewing by attracting visual attention in a bottom-up way [71]. As a framework of low-level vision processing, saliency detection facilitates subsequent processing, such as object detection or matching, by reducing computational complexity - which is a key consideration in real-time applications.

The human visual system has a remarkable ability to attend automatically only to salient locations in both static and dynamic scenes, and the computational visual attention system aims to achieve similar effects. Most saliency detection systems have a similar structure, as shown in Figure 1.12. The main idea of computing several features in parallel is to fuse their values in a single representation, which is usually called a *saliency map*. The input image is decomposed into several pre-attentive feature channels, such as colour, intensity and orientation, which operate individually over the entire visual scene. (Detailed information on how to implement such a system may be found in [61], which presents a classic saliency detection model.)

1.6 Aims and objectives

In computer vision tasks, finding salient regions in the visual field is also essential. Because it allows computer vision systems to process a flood of visual information and allocate limited resources to relatively small but interesting regions, or to a few objects, most saliency-based approaches have focused on the extraction of image locations. As we mentioned above, saliency detectors have been widely adopted in applications such as object tracking and recognition. However, this vision approach frequently lacks a formal justification. In other words, it is difficult to evaluate the proposed methods due to the absence of clearly defined optimality criteria.

Research over the last decade has built a solid computational visual saliency model for representation and analysis of 2D images. The rising demand of saliency-based applications for 3D content increases the importance of computationally modeling 3D

FIGURE 1.11: Examples of computational visual attention on 2D images [2]. Top row: original images. Bottom row: detected saliency.

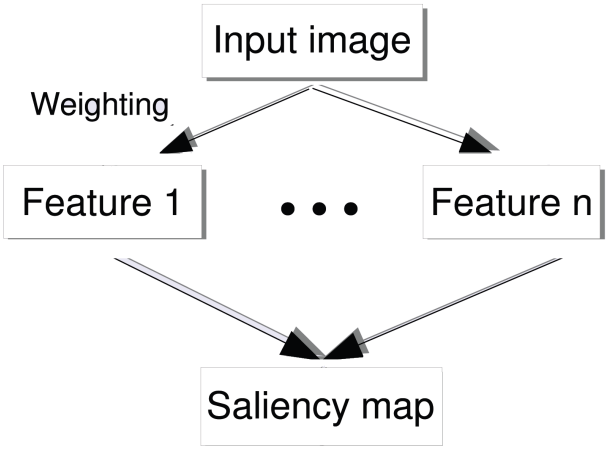


FIGURE 1.12: The main structure of the computational attention systems.

visual attention. However, extending the existing techniques of saliency detection for 3D surfaces to operate directly on large point sets is not trivial. This is so, not only due to the large size of the data, but also due to the absence of topological information regarding the point connectivity. Two main challenges need to be addressed when developing a model for 3D saliency detection: (1) the influence of 2D visual features and (2) the influence of depth on visual attention deployment in 3D viewing condition. The first question concerns the possibility of adapting existing 2D saliency models to 3D cases; the second question concerns the means by which depth information can be taken into account.

In our work, the notion of saliency is expanded to the 3D domain. This dissertation proposes several novel methods to determine the salient regions of 3D meshes and evaluate them in multiple aspects. However, a complete saliency detection method may require multiple frameworks. We therefore subdivide the main goal of this dissertation into a set of objectives, described below:

1. **Surface preprocessing: smoothing**. The scanned surface usually contains unfavourable regions or boundaries due to low reflection or specular reflection, occlusion and depth discontinuity. It is important to remove or refine the non-ideal regions in order to achieve more accurate results for subsequent applications.

2. **Feature channels generation**. Most saliency detection systems require one or more features from the 2D image or 3D shape to generate the saliency, such as the color, intensity, orientation, curvature, or shape index. We propose two feature channels: relative distance (RD), and Retinex-based importance feature (RIF), which are based on the surface geometry and the human visual perception characteristics of the object surface respectively. As a measure, RD may be used to describe the local type of shape as a continuous parameter, and has the ability to present concave and convex region significantly (see chapter 5). RIF combines the information on human visual perception with that on the surface geometry, and creates a more comfortable surface representation to illustrate the local details to the human eye (see Chapter 3). These two feature maps are combined into a unique saliency map by non-linear normalization (see chapter 5).

3. **Surface saliency detection**. We propose a saliency detection approach that considers both local and global shape information. In other words, such an approach considers not only the individual vertices, but also local patches based on groups of vertices. This method usually produces a complete salient region, while a vertex-based saliency method usually generates discontinuous salient fragments.

4. **Saliency-guided applications**: simplification, and interest points detection. Saliency detection on a 3D surface is potentially useful in numerous applications. This application is based on the idea that a vertex with a higher saliency value resides on a perceptually more interesting region of the mesh. Thus, the goal of such a saliency-guided application is to delay the simplification of the salient parts of the surface, while strengthening simplification on the non-salient regions. Interest points detection is a very helpful process in capturing the property of a point or region on a surface. We propose a more robust interest points detection method under different surface transformations, such as rotation and noise corruption.

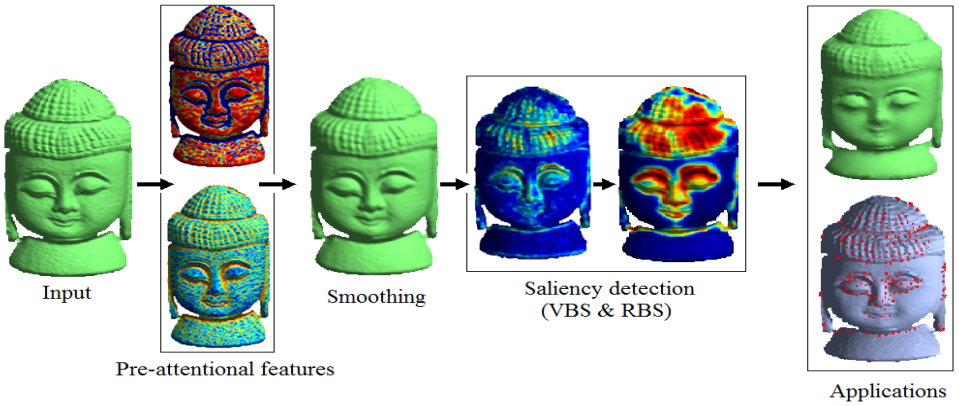


FIGURE 1.13: Overview of our technique for computing saliency on the surface, and also its applications: simplification and interest points detection.

1.7 Project Overview & Chapter Organization

Computation of surface saliency proceeds in our system as shown in Figure 1.13. Given a 3D model database, we first generate Retinex and relative distance measures for the given model as the pre-attentional features. Next, in order to improve the surface appearance - i.e., remove the scanning noise - a non-local means filtering approach is adopted. The saliency detection step consists of two types of saliency detection: vertex-based saliency (VBS) and region-based saliency (RBS), which together have the ability to provide locally and globally information-based saliency. Finally, two saliency-guided applications are demonstrated. They show how the detected saliency may be used directly to improve the efficiency and accuracy of mesh simplification and interest point detection.

The remaining chapters of this dissertation are organised as follows:

- In Chapter 2, the background of the saliency detection approach is introduced, and a literature review of existing approaches to 2D saliency detection and 3D feature points and salient region detection are provided. Moreover, as both surface smoothing and segmentation will be part of the framework for saliency detection, they will be briefly reviewed. Basic literature reviews on simplification and interest point detection are also presented.

- In Chapter 3, as a key part of the proposed saliency detection system, we present a brief review of the most frequently used Retinex methods in the 2D image context, in which the core concepts of the various approaches are introduced and their main advantages and disadvantages are discussed. On the basis of this, the application

of the Retinex theory to 3D surfaces is proposed, and we term this new feature RIF.

- In Chapter 4, a non-local means filter approach is adopted for the 3D surface. The outcome of this application is demonstrated to offer a significant improvement in the representation of the surface appearance.
- In Chapter 5, two novel saliency detection methods are proposed: vertex-based saliency (VBS) and region-based saliency (RBS).
- In Chapter 6, a saliency-guided 3D application is presented: simplification.
- In Chapter 7, a second saliency-guided 3D application is demonstrated: interest points detection.
- In Chapter 8, we draw relevant the conclusions and summarise our contributions.

Chapter 2

Literature review

As an increasing amount of 3D data has become available, and the complexity of these models has increased, several different techniques have been developed to detect salient regions on 3D mesh surfaces. In this chapter, the existing 2D and 3D image saliency detection methods will be reviewed. As discussed in Chapter 1, the proposed saliency detection method is a hybrid system, which involves smoothing and segmentation; therefore, both surface smoothing and segmentation methods will be briefly reviewed. Further, existing techniques of mesh simplification and interest points detection are mentioned as the effectiveness of the proposed saliency detection method is validated by both measures.

2.1 Saliency in 2D images

The human visual system has been an inspiration to the computer vision, robotics and computer graphics fields, in their aspiration to develop methods of detecting the most relevant parts within a large amount of visual data. The objective of human visual system-inspired models is to improve artificial vision systems by computing a numerical value for the likelihood of attending to - the *saliency* of - every location in a given image [72]. There are many existing saliency detection approaches for 2D images, and they have a similar structure: computing several features in parallel and then fusing their values in a representation which is usually called a *saliency map*.

The most general model of saliency detection appears in [61] by Itti and Koch. It consists of the following steps. First, the model computes one, or several image pyramids from the input image in order to enable the computation of features at different scales. Several feature types are commonly used, such as intensity, color, and orientation. Then

the center-surround operation compares the average value of a given central region to the average value of its surrounding region. Finally, the conspicuity maps are generated by summing up the feature maps: the conspicuity maps are then normalized, weighted and combined to form the saliency map. The saliency map is usually represented by a gray image, in which the brightness of a pixel is proportional to its saliency. Many methods have been based on such a saliency model, but in each case with a different method of fusing the feature maps, as it is not clear how the human brain maps and fuses the relative importance of features. Usually, a weighting function is applied to each map before summing up the maps within a computational system. The weight function determines the relative importance of each feature.

According to the feature determination methods, the existing saliency detection methods can be divided into four classes:

- Pixel-based methods [61, 63, 69, 73, 74, 75, 76];
- Region-based methods [62, 77, 78, 79];
- Frequency-based methods [80, 81, 82, 83] ;
- Parameter learning-based methods [2, 68, 84].

Pixel-based methods process individual pixels at single or multiple scales to obtain the feature contrast, and then use this to estimate the saliency map. Region-based methods perform computation on clustered regions or segments to obtain regional information for a saliency map. Frequency-based methods perform calculations in the frequency domain: this data is then reverted to the spatial domain to form the saliency map. Finally, the parameter learning-based method is a non-parametric approach, which learns the free parameters of the saliency model using machine learning. These four classes of saliency-detection techniques will now be considered individually.

2.1.1 Pixel-based methods

The first pixel-based method was introduced by Koch and Ullman [73]. They employed a winner-take-all approach: it demonstrates that the selective routing of information from the early representation to the central one can be performed by two complementary cellular networks. A saliency map is computed to indicate the locations of salient areas. Distinctive features, such as luminous color, or high velocity motion will pop out automatically in a stage which is called $pre-attentive$, and then the salient areas become the object candidates.

Clark and Ferrier [74] were among the first to implement an attention system based on this Koch-Ullman model. Another early model was introduced by Milanese [75]. This work introduced concepts such as conspicuity maps and feature computations based on center-surround mechanisms.

The classic pixel-based saliency model is the hierarchical model proposed by Itti et al. [61]. This method uses gradient, color and orientation information as the feature channels to simulate attention span and spread across the image. This method is highly parametric, involving the computation of a linear combination of weights to fuse the multiple feature channels. Twelve maps of red-green and blue-yellow opponent colors are computed at six different scales. These maps are then normalized, and a center-surround operation is performed using across-scale subtraction between two maps at the center and the surrounding levels. The final saliency map is computed by linear summation of the preceding conspicuity maps. This method has been built into a system called the Neuromorphic Vision C++ Toolkit (NVT).

With the proliferation of eye-tracking data, a number of researchers have recently attempted to address the question of what attracts human visual attention by being more mathematically and statistically precise. Bruce and Tsotsos [69] presented a novel model for visual saliency computation based on information-theoretic formulation approach *Attention based on Information Maximization (AIM)*. They modeled bottom-up saliency as the maximum information sampled from an image. They measured saliency at a given pixel in the image by Shannon's self-information of that location with respect to its surrounding context. To estimate the probability density of a visual feature in the high-dimensional space, they employed a representation based on independent components, which are determined from natural scenes. Moreover, Bruce and Tsotsos [85] used an Independent Component Analysis (ICA) to represent the probability distribution of local image patches using a large database of patches drawn from natural images.

Gao and Vasconcelos [63] proposed a unified model for top-down and bottom-up saliency as a classification problem. A set of features were selected to best discriminate the class of interest (e.g., faces or cars) from all other stimuli, and saliency was defined as the weighted sum of feature responses for the set of features that are salient for that class, which is similar to the concept of *Object Bank* [86]. This forms a definition that is inherently top-down and goal-directed, as saliency is defined for a particular class. Such a visual saliency model is assigned to useful locations for the task of classification. Gao and Vasconcelos [87] defined bottom-up saliency using the idea that locations are salient if they differ greatly from their surroundings. Two filters were used to measure the saliency of a point: a difference of Gaussians (DoG) filter, and a Gabor filter. The saliencies are measured as the Kullback-Leibler (KL) divergence between the histogram

of filter responses at the point and the histogram of filter responses in the surrounding region.

Walther and Koch [88] created SaliencyToolBox (STB): describing proto-objects as volatile units of visual information that can be bound into a coherent and stable object when accessed by focused attention. In addition, they extended Itti's model by a process of inferring the extent of a proto-object at the attended location from the maps that are used to compute the saliency map. However, the high computational cost and the choice of parameters are still the weaknesses of this model.

Zhang et al. [89] proposed a model that they termed Saliency Using Natural statistics (SUN). This model combines top-down and bottom-up information to predict eye movements during real-world image search model: SUN implements target features as the top-down component. In this model, saliency is computed locally, which is consistent with the neuroanatomy of the early visual system and results in an efficient algorithm with few free parameters. This measure of saliency is based on natural image statistics, rather than on a single test image.

Le Meur et al. [90] adapted the Koch-Ullman model to include the features of contrast sensitivity functions, perceptual decomposition, visual masking and center-surround interaction. Colour contrast was applied by computing the colour distance in Krauskop's color space [91]. The intra- and intervisual masking of the opponent colors of black-white, red-green, and blue-yellow are used to compute the feature maps. Then the color maps are reinforced chromatically before, passing through a center-surround interaction to select relevant visual regions. By summing the resulting achromatic channels, the final saliency map is obtained.

Ma and Zhang [92] generated a saliency map using local contrast analysis, and extracted salient objects from the map using a fuzzy growing method. This method outputs three-level mapping of attention, including attended view, attended areas, and attended points. The attended view is analogous to the direct perceptions of stimulus; the attended areas mapping simulates the early selection of human perception, in this method using a fuzzy growing method to extract attended areas or objects from the saliency map; and the attended points mapping is analogous to the results of late selection of human perception.

Harel et al. [76] exploited the power, structure and parallel nature of graph algorithms to achieve efficient saliency computations in their Graph Based Visual Saliency (GBVS) model, which is based on the use of a dissimilarity metric. This model consists of two steps: first forming activation maps on certain feature channels, and then normalizing them in a way that highlights conspicuity and admits combination with other maps.

This work defines Markov chains over various image maps, and treats the equilibrium distribution over map location as saliency values.

It can be seen that the main focus of the above methods is on the extraction of multiple low-level visual features such as intensity, color, orientation, texture and motion from the image at multiple scales. After a saliency map is computed for each of the features, these maps are normalized and combined in a linear or non-linear fashion into a master saliency map that represents the saliency of each pixel. Sometimes specific locations are identified through a combination of winner-take-all and inhibition-of-return operations.

The pixel-based saliency detection has the following advantages and disadvantages:

Advantages:

- Easy to implement, as the pixel-based methods often operate on intensity and colour features. In computing these features, simple difference between pixel values would suffice [61, 73, 90, 92].
- The models perform well qualitatively.

Disadvantages:

- The estimated saliency maps by most of the methods have relative low resolutions [61, 73, 76, 92].
- May generate saliency maps that have falsely-defined object boundaries [61, 73, 76], limiting their usefulness in certain applications. Some methods, such as [93], for efficiency it use only luminance information, thus ignoring distinctiveness clues in other channels. Pixel-based method may fail to highlight the entire salient region, or highlight the salient region partially, and frequently do not match actual human saccades from eye-tracking data [72].
- This approach requires an extensive amount of iteration, as some of the feature maps are computed using filtering in the form of distance between a pixel and its neighbors at different scales. Moreover, many models complement or supplement one feature-based detection method with another, resulting in a better detection performance but suffering in terms of algorithm complexity [61, 72, 73, 76].

2.1.2 Region-based methods

The region-based model is another fundamental saliency model. It combines some pixels into local regions, and then performs computation on clustered regions to obtain the regional information for the feature map.

Chen et al. [77] proposed a saliency region detection technique based on two types of contrasts: histogram-based contrast and region-based contrast. The former contrast-based saliency is based purely on the degree of color separation from all other image pixels. The latter, contrast-based saliency improves on the saliency results of the histogram-based contrast saliency. In order to reduce the color quantization artifact, a color space smoothing algorithm is used to improve performance. In addition, this method is also extended to the region-based contrast, which requires a segmentation procedure, and the color contrast is computed at the regional level. The saliency for each region is then defined as the weighted sum of the given region's contrasts to all other regions in the image.

Aziz et al. [78] used a region-based method to compute color, eccentricity, and orientation. The model takes a set of regions, itself consisting of several regions, as the input. The image is first segmented into regions using a mean shift algorithm. The color feature is obtained by clustering the color of pixels in each region. The saliency of a region can then be found by summing the Euclidean distance of the given region from all the other regions. A regional parameter is used to allow moderate sized regions to have a high value gain while small and large regions relating to noise and background respectively are assigned a low value gain. The color contrast is computed as the mean gradient along its boundary to the neighbor regions in the MTM (Mathematical Transformation to Munsell) color space [94]. The eccentric and the orientation feature maps are calculated using the moments of the regions.

Avraham and Lindenbaum [79] proposed an Esaliency, which stands for *extended saliency*. This model uses a validated stochastic model to estimate the probability that an image region is meaningful. It refers to this probability as saliency, and thus specifies saliency in a mathematically well-defined sense. This model too is region-based. It begins with a rough pre-attentive grouping process. The uniform regions that are suggested by the grouping process are used as initial candidates for attention. Then the judgment as to whether some part of the image is salient is context-sensitive and global. Finally, the uniqueness assumption is replaced by a preference for a small number of similar or dissimilar salient regions that may be located near to, or far away from one another.

Achanta et al. [62] proposed a frequency-tuned saliency detection method. The saliency is estimated by calculating the Euclidean distance between the arithmetic mean pixel

value of the image and the Gaussian blurred version of the original image. In this work, the authors also performed a frequency-domain analysis on five state-of-the-art saliency methods [61, 62, 76, 80, 92], obtained five advantages of this method: emphasize the largest salient objects; Uniformly highlight whole salient regions; Establish well-defined boundaries of salient objects; Disregard high frequencies arising from texture, noise and blocking artifacts; Efficiently output full resolution saliency maps.

Region-based methods have the following advantages and disadvantages.

Advantages:

- Provide good regional localization of objects [62, 77, 78, 79].
- Fill-in the gap between the human capabilities and the limited resources available on mobile robots by proposing a region-based solution for the artificial visual attention and bringing attention in harmony with the rest of the machine vision [78].

Disadvantages:

- This approach selects objects with clean boundaries, but they are often not the salient objects. This is because most of these methods were originally designed for image segmentation [62, 77]. Moreover, this method is not robust against noise, resulting in an average detection performance. In [62], it only considers first order average color, which can be insufficient to analyze complex variations common in natural images.
- High computational complexity, and high memory requirements, due to global similarities have to be estimated [62, 78, 79].

2.1.3 Frequency-based methods

Frequency-based methods examine the information content used in the creation of the saliency maps from the frequency domain. It requires that the input images be converted to the frequency domain, and subsequently that after the computation has been performed, the results are converted back to the spatial domain.

Hou and Zhang's Spectral Residual algorithm [80] focuses on exploring the properties of the background by exploiting the power of the log spectrum. This method is able to detect salient regions in visual inputs by capturing the noise that appears on the log intensity frequency curve. Given the amplitude $A(f)$ of the averaged Fourier spectrum of an image, the log spectrum representation of the image is first computed using the

following: $L(f) = log(A(f))$. Then the spectral residue $R(f)$ may be obtained by the simplification equation $R(f) = L(f) - A(f)$. The content of the residue spectrum may be interoperated as the conspicuous portion of the input image. Finally, the saliency map in the spatial domain is obtained by using Inverse Fourier Transform on the residue spectrum. The spectral residual approach does not rely on parameters setting and detects saliency rapidly.

Wang et al. [81] built a two-stage approach based on Hou and Zhang's method [80]. In the first stage, the method quickly locates the visual pop-outs (appear suddenly) from the entire image based on the spectrum residual model [80]. In the second stage, based on local coherence, the similarity and continuity results are considered to propagate the result from the first stage, to capture some details that are missed in the first stage.

Guo et al. [82] argued that the phase spectrum is key to calculating the saliency map, and proposed a model called *phase spectrum of Fourier transform*(PFT) for saliency detection. Later, Guo et al. [95] also proposed a novel multi-resolution spatiotemporal saliency detection model termed the *phase spectrum of quaternion Fourier transform* (PQFT) to calculate the spatiotemporal saliency map of an image by its quaternion representation. This method allows the phase spectrum to represent spatiotemporal saliency in order to perform attention selection not only for still images but also for videos.

Cui et al. [83] proposed making use of the temporal spectral residual on video slices, which can automatically separate foreground moving objects from their background. It is computationally efficient, being based in Fourier spectrum analysis. The basic idea is to detect saliency motion patterns by calculating the temporal spectral residual in the temporal domain, by means of a global threshold selection scheme, and a saliency majority voting operation. The authors combined the spectral residual for bottom-up analysis with features capturing similarity.

Advantages:

- Provide good detection performance (better than competing pixel-based methods). Usually detects the salient region completely [80, 81, 82, 83].

Disadvantages:

- High computational complexity, and high memory requirements, because this method would require transformation to the frequency domain. Computations are done in the frequency domain and then the results will be converted back to the spatial domain.

- Amplifies the intensity of the areas with less periodicity or less homogeneity and suppresses the intensity of the areas with more periodicity or more homogeneity, as a consequence of employing an inverse Fourier transform [96].
- This approach cannot detect smooth-texture salient regions against the complex-texture background.
- The method may falsely detect non-salient areas as salient, due to the lack of consideration given to global characteristics of the image [96].

2.1.4 Parameter learning-based methods

Most of the saliency computational models use a set of biologically plausible linear filters, such as, Gabor or DoG filters. These require many design parameters, such as number, type, and size of the front-end filters, as well as the choice of nonlinearities, weighting and normalization schemes. Kienzle et al. [68] proposed to derive a visual saliency model directly from human eye tracking data using a *support vector machine* (SVM). The model consists of a nonlinear mapping from an image patch to a real value, trained to yield positive outputs on fixated, and negative outputs on randomly selected image patches.

Seo and Milanfar [84] presented a unified framework for both static and space-time saliency detection. Initially, the method uses local regression kernels as features, which differ fundamentally from conventional filter responses, but capture the underlying local structure of the data exceedingly well, even in the presence of significant distortions. Then a local self-resemblance measurement is provided, which is a non-parametric kernel density estimation for features, indicating the likelihood of saliency. Finally, a simple but powerful unified framework for both static and space-time saliency detection is proposed.

Goferman et al. [2] presented a context-aware saliency detection method, which aims at detecting the image regions that represent the scene and not just the most salient object. The underlying idea is that salient regions are distinctive with respect to both their local and global surroundings. In addition, the authors follow four basic principles of human attention: local low-level considerations; global considerations; visual organization rules; and high-level factors. Hence, in addition to low-level features such as contrast and color, they also consider global effects which suppress frequently occurring objects, they propose that visual forms may possess several centers of gravity, and they include detectors specifically for human faces.

Parameter learning-based methods based on Gabor or DoG filter responses require many design parameters, such as the number of filters, type of filters, choice of the nonlinearities, and a proper normalization scheme. These methods tend to emphasize textured areas as being salient, regardless of context.

The parameter learning-based methods have the following advantages and disadvantages.

Advantages:

- Enables to automatically choose features relevant to visual saliency by learning specific feature weights in the integration step.
- The models perform well qualitatively.

Disadvantages:

- The models neglect some key characteristics of the HVS, such as the human visual sensitivity change due to foveation [68].
- Requires many design parameters, such as the number of filters, type of filters, choice of the nonlinearities, and a proper normalization scheme [68, 84].

2.2 3D saliency detection

Owing to its efficiency of visual persuasion in traditional art and technical illustrations, the concept of visual saliency has now been widely used in computer vision applications. More recently, the notion of saliency has been exploited in the 3D domain. The detection of salient regions has been an important pre-processing step for the analysis of mesh surfaces. It is helpful to capture the property of a point or region on a 3D surface in the interest of human perception. It can efficiently and effectively reflect the relative perceptual importance of different regions.

In general, visual saliency measures which regions or points of a 3D shape stand out with respect to their neighbors. Recently, surface saliency has been applied to a wide range of applications in the fields of computer vision and graphics, such as mesh simplification [3, 40, 47, 49, 77, 97, 98], scene rendering [49, 99, 100], view point selection [3, 49, 50, 101], point cloud matching [40, 102, 103, 104], representation [49, 105, 106, 107] and enhancement [106, 108]. In this section, we cover the most relevant work, focusing on methods designed for feature points detection and saliency estimation.

Salient points, also referred to as feature points, interest points, or keypoints, are those points that are distinctive in their locality and stable in all instances of an object, or of its category of objects [109]. Work on 3D surface saliency detection has been much less intensive than on 2D image saliency detection works, and proposed techniques may be grouped into three categories, based on the method of capturing and analysing the features of interest: local measurement-based saliency, global measurement-based saliency and eye tracking-based saliency.

2.2.1 Local measurement-based Saliency

Local measurement based saliency is a strategy of computing the saliency by considering local similarities. One of the earliest saliency estimations on 3D surfaces was proposed by Lee et al. [3]: it was inspired by earlier work on saliency detection in 2D images [61]. The authors introduced a novel approach: *mesh saliency*, which defined each vertex as a function of the differences of Gaussian-weighted mean curvatures at successive scale. By filtering the mean curvature with Gaussian filters of varying standard deviation, the various scales are obtained. Five scales are then used: $\sigma_i \in \{\varepsilon, 2\varepsilon, 3\varepsilon, 4\varepsilon, 5\varepsilon, 6\varepsilon\}$, where ε is 0.3% of the length of the diagonal of the bounding box of the model. The final mesh saliency is computed by adding the saliency maps at each of the five scales, after applying a non-linear normalization of suppression.

Gal et al. [105] have constructed salient geometric features by clustering a set of local descriptors that have a high curvature and a high variance of curvature values for partial shape matching. For a given 3D mesh, the authors analyse its geometric properties and define a sparse representation of the mesh with local shape descriptors. The key idea is that a small set of descriptors can represent a shape, provided that each descriptor effectively represents the local surface region around it. Quadric fitting is used to locally approximate each vertex. For each quadric patch, a representative point is selected, and the differential properties of the patch at that point are then calculated analytically. This method then defines a local surface patch, from which the authors employ a region-growing technique to iteratively define the local patches, in order to generate a rather small and yet effective set of local surface descriptors. The salient geometric features are then constructed by clustering together a set of descriptors that are sufficiently interesting, in the sense that they have a high curvature relative to their surroundings, and a high variance of curvature values.

The mesh saliency metric of [3] was modified by Liu et al. [107], with Morese theory. They employ a bilateral filter instead of the Gaussian filter, as they found that the Gaussian-weighted difference of fine and coarse scales can result in the same saliency

values for two opposite and symmetric vertices, due to the absolute difference. A further consideration was that it is difficult to control the number of critical points if combining saliency maps at different scales. The saliency of a vertex is defined by Liu et al. as the Gaussian-weighted average of the scalar function difference between the neighboring vertices and the vertex itself.

Novotni et al. [102] detected local salient points on a 3D voxel grid. A local spherical harmonics descriptor was computed for each salient point. Given two objects, this technique establishes correspondences between the descriptors, and determines the similarity with respect to the spatial arrangement of corresponding points and descriptor similarities. The selected points are found as local extrema of the differences of Gaussian filters applied to the characteristic function of the surface.

Tan and Chau [98] proposed to improve image-driven simplification by modifying the error metric to correlate better with perceptual difference. They adapted Itti's method for detecting salient parts from a given mesh: extracting two sets of feature maps, which indicate intensity features and orientation features, normalizing each map, and then summing all maps to create the saliency map. The intensity feature maps are computed by applying a center-surround operator at different scales on the input image. Gabor filters are then applied at four different angles to generate the orientation feature maps.

Shilane and Funkhouser [40] analyzed distinctive regions based on performing a shape-based search using each region as a query into a database. The distinctive regions have shapes consistent with objects of the same type, and different from objects of other types. The authors generate for each mesh a set of random points that are the centers of spherical regions covering its surface at multiple scales. The shape of every spherical region is then described by an *Harmonic Shape Descriptor* (HSD) with 32 spherical shells and 16 harmonic frequencies. The shape descriptor offers fast computation, is compact for storage, is indexable, and allows for rapid comparison. The distinctive properties of every shape descriptor are then compared to the contents of a database containing multiple classes of objects. The distance of one of the features in the given model to the closest descriptor is given, and the distinctiveness of the descriptor is computed by evaluating a retrieval performance metric: *discounted cumulative gain* (DCG). Finally, the computed measure of class distinction needs to be mapped back onto the vertices of the mesh by modelling distinction as a mixture of Gaussians.

Castellani et al. [103] generated the salient points for sparse points matching. In this work, the detected mesh saliency is used to combine with statistical descriptors. The first step is applying Gaussian filters to the given mesh, and using a *Difference-of-Gaussian* (DoG) operator to determine the multidimensional filtering maps. A scale map is obtained, in order to reduce variation of displacement in the resulting scalar quantities. In

general, the most significant motion of the vertices is along the direction perpendicular to their local surface. Therefore, the authors project the vector to the normal of the vertex. Moreover, each map is normalized by adopting Itti's approach [61]. The salient points are detected by an adaptive inhibition-process on each normalized scale map. If the value on the scale map observed in the neighborhood of v are higher than 85% of the values in its neighborhood, the value is retained: otherwise, set to zero. Furthermore, the authors use the *Hiddern Markov Model* (HMM) to build a compact description able to summarize information related to interest points and to their neighborhood area. A HMM-based matching is given to detect links between similar points in two different views' of the same object. It is formally cast as a *maximum weighted matching problem* (MWMP) after considering the similarity measures as weights that characterize links between points.

Local measurement based saliency detection methods have the following advantages and disadvantages.

Advantages:

- Easy to implement; low memory requirements. The local measurement-based methods do not require to calculate the similarity of the features through the whole models, and the features are often curvature-based[3, 40, 105].
- The models perform well qualitatively.

Disadvantages:

- The estimated saliency maps generated by most of these methods display discontinuous salient regions, sometimes may completely fail to highlight the salient region, or highlight the salient region partially, as only local variations are considered during the saliency estimation [3, 40, 102, 105], but the global geometry is also a very important to factor to obtain a faithful saliency map, which is more close to simulate the human visual attention [50].
- Produces unstable results under input data transformations, such as noise corruption, viewpoint changing, and scaling. Lee's mesh saliency [3] computation is a geometry filter (smoothing) operation in terms of the mean curvature used with the center-surround mechanism, however, the curvature is very sensitive the noise. It might make the same saliency for two opposite and symmetric vertices because of using the absolute difference between the Gaussian-weighted average [110].

2.2.2 Global measurement-based Saliency

Global measurement based saliency is an approach to estimating the saliency from the whole 3D model. There have been far fewer methods based on this strategy than on local-based saliency.

Feixas et al. [101] presented a unified framework for viewpoint selection via view-based mesh saliency. The definition of view-based mesh saliency is derived from the dissimilarity between two polygons. The polygonal dissimilarity between the set of input polygons and a set of viewpoints is estimated by using the Jensen-Shannon (JS) divergence between the cost of computing the probability distributions. The greater the similarity between two polygons, the smaller the JS-divergence. Polygonal saliency is defined by degree of dissimilarity. A polygon will be salient if the average of JS-divergences between itself and its neighbors is high. This method also estimates viewpoint saliency, which can transfer the polygon saliency to the sphere of viewpoints, using the conditional probabilities of the reverse channel. In a similar manner to Lee's Mesh saliency approach, this method is also used to select the best views.

Song et al. [111] introduced a saliency detection method that incorporates a *Conditional Random Field* (CRF) framework. A CRF is designed to robustly detect salient regions utilising neighbourhood consistency. More recently, the author produced a mesh saliency detection via spectral shape analysis [112]. The author claimed that spectral analysis can provide information about global cues or statistics of images which are difficult to extract in the spatial domain. The saliency map is constructed by a means of curvature-weighted heat diffusion function to deliver and reorganise the saliency information into the spatial domain in a multi-scale way.

Leifman et al. [50] proposed a viewpoint selection method based on vertex distinctness. They introduced a diffusion distance-based dissimilarity measure, which is robust to small changes in the mesh. This dissimilarity measurement models the difference between the two histograms as a temperature field, and considers the diffusion process on the field. The dissimilarity between two vertices is calculated as proportional to the diffusion distance and inversely proportional to the geodesic distance. Finally, saliency is obtained by the single-scale distinctness of each vertex.

Wu et al. [113] proposed a method to detect the mesh saliency based on the observation that salient regions are both locally prominent and globally rare. A multi-scale local descriptor that can capture local geometric feature with different sizes of regions was introduced for the measurement of local contrast. The clustering approach was used in order to reduce the required computation. Finally, the mesh saliency is obtained by the linear combination of the local contrast and the global rarity.

Global measurement based saliency detection methods have the following advantages and disadvantages.

Advantages:

- Provide good detection performance. For example, in [113], the global saliency of each vertex, based on its contrast with all other vertices, was defined as the measure of global rarity. This enables geometrically similar vertices to have comparable saliency values.
- More robust to noise and object rotation than local measurement-based saliency, as this measurement considers the global and local features of the given model rather than only considers local details [50, 101, 111].

Disadvantages:

- High computational complexity, and high memory requirements. Because this approach estimates the saliency through the whole 3D model. For example, it requires several procedures to estimate the surface saliency in [50]: vertex descriptor estimation, dissimilarity measurement, distinctness computation, and multi-scale computation. Such multiple prcocessing increase the computational complexity rapidly.
- Some of the methods [101, 113] combine the local and global shape descriptor, and they may require to define a multi-scale manner to determine the local shape descriptor, it is well-known that such multi-scale manner is insensitive to a certain amount of noise on the mesh surface [3].

2.2.3 Eye Tracking-based Saliency

Eye tracking-based saliency is a method based on eye-tracking equipments. The saliency was obtained by tracking the eye movements through an eye-tracking-based user study.

Howlett et al. [97] introduced a saliency prediction for simplified polygonal models. The authors attempted to determine salient features by using eye-tracking devices. The visual fidelity of simplified polygonal models can be improved by emphasizing the detail of salient features identified through capturing human gaze data. The location of a participant when fixating while viewing a particular model was captured by an SMI EyeLink high-speed eye-tracking system. The saccade instants are detected by measuring the difference between the current eye position and the average of the last six eye positions.

The saccade case will be recorded if the size of the visual angle was greater than some specified threshold value. The system then keeps tracking the faces in the polygonal model on which the participant's eyes focused since the last saccade, until a new saccade was detected, at which point the information is updated with the fixation data. The information on fixations was summed over all participants to give us the final data for each object. Finally, the obtained perceptually important features are incorporated into a simplification method and used to evaluate the visual fidelity of each of these models.

Kim et al. [106] proposed a saliency-guided enhancement for volume visualization. They assumed that a saliency value is assigned to each voxel of the volume data. A saliency field was computed using Lee's idea [3] on a virtual emphasis field. The emphasis field was then used to specify multi-scale enhancement, achieving it through modulation of various volumetric aspects of appearance such as color and opacity. Moreover, Kim et al. [108] presented a user study that compared earlier mesh saliency approaches based on Lee's methods with the saliency which was detected based on the human eye movements. The normalized, chance-adjusted saliency was introduced by improving the previous chance-adjusted saliency measure, in order to quantify the correlation between mesh saliency and fixation location for 3D rendered images. The authors hypothesized that the computational model of mesh saliency has better correlation with human eye fixations than a random model and a curvature-based model for the first few seconds after stimulus onset. They then used an ISCAN ETL-500 monocular eye-tracker to capture the eye movement of participants. In addition, a calibration step was used to obtain an accurate correspondence between the eye-tracker space and the monitor space for that participant. The second step was that the captured data was grouped into two types: fixations; and saccades. The authors then took into consideration the normalized chance-adjusted saliency, inspired by Parkhurst et al. [114].

A Schelling points detection method for 3D meshes has been proposed by Chen et al. [77] recently. The general idea of this approach is to gather a large collection of feature points from individuals, and then to study what geometric properties distinguish them from others. Specifically, a large number of Schelling points are acquired by asking people to select points on 3D surface meshes that they expect will be selected by other people. It is difficult for the purposes of this model to collect enough data to analyze, because it requires recruiting and supervising many human subjects in a user study. To this end, the author performed the study on-line. The Schelling data is generated by the following procedures: data filtering, mesh selection, protocol implementation, data collection and Schelling point extraction. The authors extracted a discrete set of Schelling points and build an indicator function from the local maxima of schelling distribution function. The next stage of this study is analysing the schelling points from multiple aspects:

consistency, symmetry, distribution and distinctiveness. The author also investigated the predication of the Schelling distributions, providing a method that estimates the probability that a person would have selected each vertex of a mesh if it had been included in the study's previous steps.

Eye-tracking saliency detection methods have the following advantages and disadvantages.

Advantages:

- The detected saliency is highly relevant to the human visual system.

Disadvantages:

- High costs on the eye-tracking systems.
- Systems require long set-up times.
- Same object may generate different salient regions, as people's visual attention are not always in agreement.

2.3 Surface Smoothing

With geometry scanners becoming more widespread and a corresponding growth in the number and complexity of scanned models, robust and efficient geometry processing becomes increasingly desirable. The quality of 3D surface models is of vital importance in many fields, such as reliable path planning and obstacle avoidance in robot navigation, and physical realism in model based 3D object recognition and computer graphics. However, the acquired data usually contains imaging noise due to low reflection or specular reflection, occlusion and depth discontinuity. Sometimes, a rough surface is generated due to the rapid change of orientations and vertex locations of reconstructed surfaces caused by noise introduced in the process of surface scanning and image registration and integration. Therefore, it is an active area of research to effectively filter noise from the reconstructed surface models, while preserving their desired details. Mesh smoothing, or denoising, is a process dedicated to the removal of noise with minimal damage caused to geometric features of the object.

A wide variety of mesh smoothing algorithms has been proposed in recent years. According to the isotropism of the smoothing kernel (by isotropic is meant that rotation of the covariate co-ordinate system will not change the result of smoothing), the smoothing methods may be classified into the following two main categories:

- Isotropic smoothing [34, 115, 116, 117, 118, 119, 120, 121, 122, 123].
- Anisotropic smoothing [124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135].

2.3.1 Isotropic Smoothing

Laplacian smoothing is the simplest and most efficient methods among isotropic methods. Laplacian smoothing has established itself as one of the most common methods among all the geometric flow-based methods, but usually leads to volume shrinkage and over-smoothing when applied to a noisy 3D surface. Therefore, many other kinds of weighted Laplacian methods have been proposed [122, 127, 136]. To perform Laplacian smoothing, in its simplest form, is to move each vertex to the arithmetic average of the neighboring points. The Laplacian smoothing flow may be considered as the gradient descent flow for a simple quadratic energy functional. The Laplacian flow moves recursively each vertex of the mesh by a displacement equal to a positive scale factor times the average of the neighboring vertices. Due to its simplicity, Laplacian smoothing is by far the most common smoothing technique.

Taubin [123] proposed a mesh smoothing method by using a simple, isotropic technique to improve the smoothness of a surface mesh. Peng et al. [115] gave a denoising algorithm for geometric data represented as a semi-regular mesh on the basis of adaptive Wiener filtering. Despite their high speed, these methods often yield significant volume shrinkage and undesired mesh distortion.

Taubin's work is extended to smooth irregular meshes by using geometric flows by Desbrun et al. [116]. They overcome the problem of shrinkage by rescaling the mesh to preserve its volume; again, however, distortion of prominent mesh features occurs.

Kim and Rossignac [117] developed a general autoregressive moving average filter approach, which is used to combine Taubin and Desburn's approaches. The filter can act as a lowpass, bandpass, highpass, notch, or as band amplification through a suitable choice of parameters, thus enabling the user to filter out high frequency noise and enhance or suppress certain other features at the same time. However, with this method it is difficult to remove noise, while preserving sharp regions.

Isotropic smoothing methods have the following advantages and disadvantages.

Advantages:

- Easily and simply implemented. It depends only on the topology of the mesh, and not the position of the vertices. For instance, Laplacian smoothing simplily moves each node to the centroid of the polygon formed by its adjacent nodes [116, 123].

- Efficient smoothing of noisy surfaces, especially for flat areas.

Disadvantages:

- May result in distorted or even inverted elements near concavities when the model contain irregular triangulations. Sometimes produces volume shrinkage, and inverted elements [123].
- This approach smooths the surface indiscriminately, and hence prominent surface details are lost after smoothing [116, 123].
- Does not always move the node to the optimal position to get the best element quality. Tends to lose element size uniformity if iterated more than a few times. Tends to yield lower quality elements if iterated more than a few times [123, 137].

2.3.2 Anisotropic smoothing

Laplacian smoothing approaches denoise the surface indiscriminately, and hence prominent surface details are lost after smoothing. The shrinkage problem may be tackled by methods utilizing spectral analysis of the mesh signal, which is the main idea behind methods of the second category. A great deal of work has been done in the area of optimization-based smoothing [119, 121]: here, the vertex is moved so as to optimize some mesh quality. The results of these methods do improve on the inadequate feature-preserving of the Laplacian smoothing approach, but they significantly increase computation costs due to their use of partial differential equations [138] or large linear systems [139].

A variety of anisotropic methods were presented for better preserving geometric features. Regularization technique is a popular smoothing approach among anisotropic smoothing. Regularization performs smoothing operations by minimizing an energy function

$$f(x) = g(x) + \lambda h(x) x \in R^3; \tag{2.1}$$

that includes a data compatibility term $g(x)$ and a smoothing term $h(x)$. Minimization of $g(x)$ involves the compatibility of the solution to the original surface, and minimization of $h(x)$ incorporates prior knowledge. λ is called the regularization parameter which determines the weight of minimization between $g(x)$ and $h(x)$.

Welch et al. [140] described an approach to designing and fairing freeform shapes represented by triangulated surfaces. Kobbelt [127, 128] proposed a general algorithm to fair a triangular mesh with arbitrary topology by estimating the curvature for the mesh

model. Most Regularization technique works considering the surface as a height map $z(x, y)$. Blake et al. [124] introduced the membrane and plate model. The membrane has intrinsic resistance to creasing. In order to fit crease discontinuities, a plate model should be used. Stevenson and Delp [125] used the heightmap as the sum square integral of the two principal curvatures of points on a surface. Yi and Chelberg [126] proposed a simple first order smoothing term because the first order models entail significantly smaller computational efforts than second order models. Sun [129, 130] proposed a new energy minimization-based smoothing method based on locally adaptive minimization of the surface area. The position of each vertex is adjusted along the surface normal to minimize the simplex area.

Optimization-based smoothing is a newer form of smoothing that has been receiving more attention lately. Instead of moving nodes based on a heuristic algorithm, as is done in Laplacian smoothing, in optimization-based smoothing the nodes are moved so as to minimize a given distortion metric. In this method, the quality of the surrounding elements of a node is measured, and an attempt is made to optimize by computing the local gradient of the element quality with respect to the node location. This optimum value is often defined as maximizing the minimum of mesh quality metrics such as the minimum interior angle of triangles and the internal angles at the internal points. Let X be the nodal location of a node: the optimization process is to find the best location in iterations takes the form

$$x^q = x^{(q-1)} + sd^q, \tag{2.2}$$

where q is the iteration number, d^q is the vector of the search direction and s is the step length to move in this search direction.

One of the first optimization-based smoothing algorithms was developed by de Cougny [131]. This technique improved on the distorted representation of tetrahedral elements. An element distortion metric is presented that is basically the scaled ratio of an element's volume to its face areas. The author also proved several properties, that indicate that the metric may be well-suited for optimization-based smoothing. Parthasarathy [141] developed an optimization-based technique for triangular and tetrahedral meshes, which solves a nonlinear, constrained, global optimization problem, using the element's aspect ratio as the objective function to be minimized. This method uses a modified version of the feasible directions algorithm to drive the optimization. Oddy's [142] distortion metric was used to develop a global optimization by Canann [132]. Recursive local optimization has proven to be more feasible. Freitag [133] developed an approach that works to maximize the minimum angle in triangular or tetrahedral meshes. A non-smooth optimization is proposed by using an analogue of the steepest descent method of smooth functions. Amenta [135] presented theoretical results showing how some local

triangle and tetrahedral shape optimizations can be solved in linear time using generalized linear programming. For the mesh smoothing problems that don't fit into that class of problems, other efficient algorithms are presented. Many distortion metrics are discussed and various optimization techniques are compared. Mezentsev [134] presented a generic approach to mesh global optimization via node movement. Mesh is considered as an electric system with lumped parameters, governed by the Kirchhoff's voltage and circuit laws. The proposed optimization technique may be also used for solving the mesh deformation problem.

Anisotropic smoothing methods have the following advantages and disadvantages.

Advantage:

- Guarantees good quality elements.
- Well preserving the surface features, such methods have the ability to smooth data in a nonlinear way, allowing the preservation of significant image discontinuities.
- Mathematically proved.

Disadvantage:

- Cannot guarantees work on an arbitrarily defined surface. Some methods may not applicable to triangular, quadrilateral and tri-quad mixed meshes in a consistent manner [140].
- Trade-off between given original surface and a bumpy coarse surface [127, 128]. Produces volume shrinkage [132, 135].
- The computational cost, however, is much higher than isotropic methods [134, 141].

2.4 Surface Segmentation

The surface segmentation method has become a core ingredient in many mesh manipulation algorithms. In our project, the segmentation approach is employed to segment the surface into sub-regions according to vertex-based saliency, which is in order in turn to generate region-based saliency (see Chapter 5). As a framework of our project, surface segmentation approaches will be reviewed in this section.

Over the last decade, many 3D mesh segmentation algorithms have been proposed. Based on their differing aims, existing mesh segmentation methods may be generally

categorized into two classes. The first class is aimed at applications such as reverse engineering of CAD models, which tries to achieve a best fit to one of a given class of mathematical surfaces. The second class aims to segment natural objects into meaningful pieces, as expected by a human observer. In addition, we have classified the major possible approximate solutions for mesh segmentation according to the approaches taken as follows: region growing [143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157]; clustering [152, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168]; and spectral analysis [107, 151, 169, 170, 171, 172]. For a recent review and comparison of the most well known mesh segmentation techniques see Attene et al. [167] or Shamir [173].

2.4.1 Region Growing

Region growing is the most popular method in the literature for mesh segmentation. The algorithm for region growing starts with a seed element and grows a sub-mesh incrementally. The local surface properties, such as principal curvatures, are always used as criterion for growing regions with similar attributes. Kalvin et al. [143] used a region growing algorithm with a set of representative planes for the cluster approximated by an ellipsoid. The distances of all face vertices are used as the clustering criteria. Chazelle et al. [144] proposed a convex decomposition of a mesh by applying region growing with a random starting face. Lavoue et al. [146] obtained a method that uses curvature as the criteria for growing constant curvature clusters.

Kraevoy et al. [145] suggested that the approximately-convex parts may be extracted from the given model by growing patches from seed triangles and measuring convexity and compactness. Eck et al. [147] created Voronoi-like patches on the mesh and then used the dual of the patches as the base triangular mesh. A method which simultaneously segments the mesh and defines a parametrization is defined by Sorkine et al. [148]. Levy et al. [174] extracted feature contours and used them as boundaries between charts to grow the region inward.

Furthermore, in [149, 150, 151, 152, 153, 155, 156], the watershed region growing algorithm is found to have the capability to segment the mesh. The seeds for growing are based on the definition of a height function on the mesh. Recently, Moumoun et al. [175] suggested the use of the watershed principle on a hierarchical transformation of connected faces structure based on the principal curvature.

Region growing-based segmentation methods have the following advantages and disadvantages:

Advantages:

- The simplest of all possible approaches.

Disadvantages:

- The major drawback in region growing is its dependence on the initial seed selection. It may sometimes create unsatisfactory global results, because this approach searchs for local optimum of each region separately. The number of regions depends heavily on the choice of initial seeds [173].
- Region growing-based methods always stop at boundaries with a high curvature: the mesh will be over-segmented by such an approach [146].
- Due to its local nature, region growing is susceptible to over-segmentation. Thus a merging step in postprocessing is often necessary [146].

2.4.2 Clustering

Clustering-based methods merge pairs of regions from the bottom to the top of the given surface, hierarchically or iteratively, searching for the best segmentation for the given number of clusters. Attene et al. [167] proposed a hierarchical clustering algorithm based on fitting primitives. Gelfand et al. [158] introduced a segmentation approach based on slippage analysis, which used hierarchical clustering to merge points into larger regions based on slippage similarity scoring. Katz and Tal [168] demonstrated an algorithm that proceeds from coarse to fine. Each node in the hierarchy tree is associated with a mesh of a particular patch, and the root is associated with the whole input object. Golovinskiy et al. [159] described a hierarchical clustering algorithm which every face of the mesh starts in its own segment. The decomposition procedure of this method uses a set of randomized minimum cuts to guide the placement of segmentation boundaries. Sheffer et al. [162] worked on the dual graph of the mesh. Duda et al. [163] proposed an iterative process based on the k-means algorithm that begins with k representatives representing k clusters. Shlafman et al. [165] described an algorithm based on K-means clustering of faces. Another variant of k-means algorithm is presented by Cohen-Steiner et al. [164] for the creation of planar shape proxies. Wu and Kobbelt [152] used the planes, spheres, cylinders and rilling ball blend patches to define the possible proxies to other surface elements. Julius et al. [166] proposed a different variation on the iterative clustering algorithm that uses quasi-developable patches as proxies. The detection mechanism is actually narrowed to a subset of developable surface.

The clustering based segmentation methods have the following advantages and disadvantages:

Advantages:

- Easily and simply implemented.

Disadvantages:

- It has been found that the clustering approach is excessively time-consuming for mesh segmentation [158, 168]. It is the necessity to compute pairwise distances, making it expensive or even prohibitive to handle large models directly [165, 168, 176].
- The technique is non-robust, due to the k-means approach, existence of bad local minima, and the difficulty of choosing an appropriate k [162, 163, 165]. For some methods, such as [177] tends to oversegment a model into more pieces than expected or desired.

2.4.3 Spectral analysis

Spectral clustering has received a great deal of attention recently in computer vision and machine learning. Shi and Malik [169] outlined a spectral graph theory that details the relationship between the combinatorial characteristics of a graph and the algebraic properties of its Laplacian. Karni [170] proposed the Laplacian matrix of the vertex adjacency graph for mesh compression. Liu and Zhang [107] used a slightly different formulation: a symmetric affinity matrix is constructed. This matrix may be viewed as the adjacency matrix of a weighted graph whose nodes are the mesh faces.

More recently, the outer contour of the 2D spectral embedding of the mesh is used to guide the segmentation by Liu [172]. Two different operators are used for the spectral projection: structural segmentability; and geometrical segmentability. Zhou et al. [155] provided an interesting observation on the properties of spectral analysis of the normalized geodesic distance matrix of vertices on the mesh. Very recently, an unsupervised segmentation of a set of shapes via descriptor-space spectral clustering has been outlined by Sidi et al. [178]. Huang et al. [179] presented a novel linear programming approach to jointly segment the shapes in a heterogeneous shape library, producing comparable results to the supervised approaches on the Princeton Segmentation Benchmark.

The spectral analysis based segmentation method have the following advantages and disadvantages:

Advantages:

- This algorithm can co-segment a set of shapes with large variability, revealing the semantic shape parts and establishing their correspondences.

Disadvantages:

- Due to the initial segmentations required for co-analysis, this algorithm may fail if the single-shape segmentation is poor.
- High computational complexity, and high memory requirements. Because this method would require transformation to the frequency domain. Computations are done in the frequency domain and then the results will be converted back to the spatial domain.

2.5 Simplification

As we mentioned in Chapter 1, simplification for a 3D model is becoming more and more necessary. The problem of approximating a given input mesh with a less complex, but geometrically faithful representation is well-established in computer graphics. In the past, many simplification methods have been developed. Approaches to simplification may be grouped into two basic categories, based on how the operator deals with the input mesh: local, and global simplification, respectively. Local strategies that iteratively simplify the mesh by the repeated application of some local operator, and global strategies that are applied to the input mesh as a whole.

2.5.1 Local simplification

Local simplification acts on a small collection of elements and produces a new mesh with fewer elements. Local simplification strategies are much more common than global strategies. Three main variations are discussed below.

2.5.1.1 Vertex Decimation

The decimation algorithm is designed to reduce isosurfaces containing million of polygons. Some techniques offer efficient processing, but produce simplified meshes which are visually undesirable [41, 152, 180, 181]. Others create approximations more worthy of consideration but require excessive and expensive processing time and are difficult to implement [182].

Normally, most of the simplification methods begin by defining a mesh operation and then apply it to a mesh, act on a small collection of its points, and produce a new mesh with fewer vertices. Schroeder et al. [41] published one of the first algorithms to simplify general polygonal models and coined the term decimation, which operates on a single vertex by deleting that vertex and re-tessellating the resulting hole: it then measures the distance from the vertex to the average plane by its adjacent triangles, then uses that distance to decide the order in which vertices are to be removed. This method is simple to implement, but generates low-quality approximated models. Wu and Kobbelt [152] used a random selection of vertices to be removed in the context of streaming to simplify large meshes.

These methods are based on the removal of vertices from the mesh. Once a vertex is removed, all faces using that vertex are also removed and then the hole is retriangulized. Because of the way it creates triangles, this kind of algorithm is limited to manifold meshes [183].

2.5.1.2 Edge Contraction

Hoppe et al. [43, 184] investigated edge contraction, which has since become the most commonly-used simplification operation. The authors use the edge-collapse operator to construct a progressive mesh, and then measure the distance from the proposed new triangles to a set of sample points from the original mesh to decide which edge to collapse. Ronfard and Rossignac [180] also use the edge-collapse operator to simplify the mesh. They associate a set of planes with each vertex and the new vertices inherit the plane equations from the surfaces of the two merged vertices when a contraction is performed. Garland and Hechbert [1] proposed a vertex pair-collapse operator, which may be considered the topology-modifying variant of the edge-collapse operator. In practice, this method of simplification probably achieves the most efficient results.

2.5.1.3 Simplification Envelopes

Cohen et al. [185] presented a completely different approach which is appearance-based: simplification envelopes. The method measures the amount of deviation caused by the operation in the screen-space representation of the mesh. The simplification envelopes of a surface consist of two offset surfaces, or copies of the surface offset no more than some distance from the original surface. Lindstrom [186] described a more general image-based simplification strategy that does not require specialized hardware or software algorithms. Zelinka and Garland [187] proposed the use of *permission grids*, which provides tight

error bounds on arbitrary triangulated meshes, while allowing topological changes during simplification.

The local simplification methods have the following advantages and disadvantages:

Advantages:

- Easily and simply implemented. They allow the user to specify the desired attributes of the target approximating mesh with a high degree of precision [188]. For example, the algorithm may be allowed to run until the mesh contains a specific number of faces.
- In simplification envelopes approach [185], the reuslts of the simplified models are nearly visually indistinguishable from the original.

Disadvantages:

- Vertex decimation may subdivided the large planar regions into many redundant [41].
- Edge Contraction alters the topology of the input mesh, repeatedly contracting all the edges around ahole will eventually close it. It may in principle be applied indiscriminately to edges containing non-manifold vertices [1, 43, 188]. In addition, the optimal target position for a given contraction is a non-linear problem and is in practice very inefficient [180].
- The input mesh is required to be an orientable manifold due to their construction, and it limits their ability to assist in drastic simplifications as this approach preserves the topology carefully, and avoidance of self-intersection [185, 186].

2.5.2 Global Simplification

The global strategies are applied to the input mesh as a whole. This approach is far from prevalent in the simplification literature. Nonetheless, they are worth reviewing in brief.

2.5.2.1 Vertex Clustering

Rossignac and Borrel [189] proposed a vertex clustering method to handle meshes of arbitrary topological structure. This method assigns a weight to each vertex on the input

mesh by its perceptual importance, then subdivides the mesh into a three-dimensional grid: and finally, all the vertices in a given grid cell are clustered to the position of the vertex with maximum weight. This method tends to be very fast but the visual appearance of the final mesh is relatively inaccurately defined.

2.5.2.2 Surface Approximation

Cohen-Steiner's variational surface approximation [164], relies on a global non-linear optimization process to find the best mesh with N polygons by subsampling the input shape. The optimization function may vary according to the application.

2.5.2.3 Feature-preserving Simplification

Lindstrom [190] employed a perceptually motivated metric in a mesh simplification algorithm. Luebke and Hallen [45] proposed a perceptually-driven mesh simplification approach that controls the simplification using psychophysical models of visual perception. The authors mapped an edge-collapse operation to the worst contrast grating introduced by the edge in question [47]. Williams et al. [48] proposed a simplification method that relies on level-of-detail (LOD) techniques. These techniques simplify the geometric representation of a scene to reduce its rendering cost, while attempting to preserve visual fidelity. The authors followed Luebke's [45] approach of equating local simplification operations to a worst-case grating. Lee et al. [3] introduced a saliency guided simplification in order to evaluate the effectiveness of their mesh saliency method. They modified the quadrics-based simplification method by weighting the quadric with mesh saliency. Howlett [97], using an eye tracker, showed that preserving high saliency areas of natural object models improves the visual fidelity of simplified objects.

Global strategy-based simplification methods have the following advantages and disadvantages:

Advantages:

- Vertex clustering is extremely efficient and simple to implement, and by choosing the resolution of the overlaid grid, the the level of simplification may be controlled [189].
- Surface approximation do well qualitatively, and preserve feature better than those which are local strategy-based simplification method [1].

- Feature-preserving-based methods are robust to noise, as the retained points are sampled based on the a perceptually motivated metric [45, 47, 48, 190]. It requires a parameterization of the mesh in order to function, thus, the coordinates of salient region must be computed before the simplification.

Disadvantages:

- Vertex clustering may drastically change the topology of the original mesh in an unpredictable manner, and in practice does not produce very faithful geometric approximations with high simplification rate [188, 189].
- The process of surface approximation is not particularly efficient. It requires to segment the input mesh into a set of non-overlapping connected regions, then fit a locally-approximating plane to each one.
- Feature-preserving-based methods are time consuming. This approach requires to estimate the feature region or salient regions before the simplification step.

2.6 Interest Point Detection in 3D

In Chapter 6, we present a saliency-guided interest point detection application. Existing approaches to interest point detection will be roughly reviewed in this section. As with saliency detection, many models have considerably large sizes; consequently, it is necessary to select the most distinctive points on a 3D model in order to maintain efficiency in the processes applied to them.

Most of the models for interest points detection methods in 3D meshes rely on local surface descriptors, such as curvature; extrema of which are assumed to correspond to candidate interest points. In order to analyze a 3D surface at successive scales to search for interest points at various levels of detail, a multi-scale approach is usually employed in practice. The most well-known example of the power of the scale-space representation approach for applications in computer vision is the *Scale Invariant Feature Transform* (SIFT) of Lowe [191]. SIFT is used for extracting keypoints in images, and also for computing local descriptor vectors, which are then used for establishing correspondences between images.

One of the main attributes of SIFT features is the scale associated with each extracted keypoint, which in turn gives rise to the invariance of the approach to scale changes between images. The earlier work in the literature focused on feature extraction, invariance of the features to geometric transformations, and their performance in 3D object

application, such as shape matching and retrieval. In this section, we have divided the proposed 3D detectors into two categories, based on the point of view of the scale space: fixed-scale detectors [151, 192, 193, 194, 195, 196]; and scale-invariant detectors [98, 103, 110, 197, 198, 199].

2.6.1 Fixed-scale detectors

Fixed-scale detectors find distinctive keypoints at a specific, constant scale, which is provided as a parameter to the algorithm. Hu and Hua [196] defined the geometric energy of a vertex as a function of the eigenvalues and eigenvectors of the Laplace-Betralmi spectrum of a given object. A point is selected as an interest point if it remains as a local maximum of the geometry energy function within several successive frequencies. Thus the distinctiveness of an interest point is required to be stable within a portion of spectrum. In addition, the energy provides the scale at which the selected vertices are deemed interesting.

Sun et al. [151] proposed a feature point detection approach based on *Heat Kernel Signature* (HKS). The *Heat Kernel Signature* is defined as a temporal domain restriction of the *Heat Kernel* on a manifold, which is related to the Laplace-Beltrami spectrum. The response of the operator is a positive semi-definite matrix of size $N \times N$, where N is the number of vertices in the 3D model. Each vertex has an associated signature. A vertex is selected as an interest point when for large time values, its signature has a maximum with respect to the neighboring vertices.

Mian et al. [193] defined a local coordinate system -which is invariant to global rotation and translation of the 3D object- around a point using the cropped surface surrounding it. The detected keypoints are highly repeatable between partial views of an object and its complete 3D model. Additionally, the keypoint detection algorithm is presented along with an automatic scale selection technique for subsequent feature extraction. The drawback of this approach is the combinatorial exploration of vertex pairs required to define a local frame, which leads to higher computational complexity.

Zhong [194] introduced a descriptor called *Intrinsic Shape Signature* (ISS) to characterize a local/semi-local region of a point cloud. The features are constructed by a 3D occupational histogram of the supporting spherical neighborhood with a specific radius.

Darom and Keller [195] proposed an intrinsic scale detection scheme per interest point and used it to derive two scale-invariant local features for mesh models. The authors presented a scale-invariant spin image local descriptor that is a scale-invariant formulation of the spin image descriptor. Thus, local features relating to corresponding points

in different manifestations of a particular mesh will be computed over a similar support, yielding scale-invariant mesh descriptors.

The 2D SUSAN operator has been extended to 3D meshes to compute the degree of saliency of the vertices by Walter et al. [200]. The pixelized window in 2D SUSAN becomes a voxelized sphere when applied to 3D meshes. By centering the voxelized sphere at a vertex, the mean saliency may be obtained by considering the isotropy of the circular analysis window, which depends on the volume of the intersection of a ball and the surface neighborhood of the inspected point.

Knopp et al. [201] built scale-space out of a voxelized version of the original mesh: 3D-SURF. This method extends a robust 2D feature descriptor, SURF, to use in the context of 3D shapes. The features are quantized and used within a Hough approach, which retains the influence of each feature. The quality measurement, computed for each grid bin and at different octaves, is the Hessian of Gaussian second order derivatives that given the nature of the data, can be computed efficiently by means of box-filtering.

Fixed-scale detectors have the following advantages and disadvantages:

Advantages:

- This approach is simple. They compute a distinctiveness, or quality, measurement associated with each point, that can be either point-wise or region-wise, i.e. a a property of a vertex of the mesh or a property of a region around each vertex. The interest points can be selected by the maximum of the local feature.
- For method [151], is invariant to isometric deformations, which can offer a wide degree of invariance.

Disadvantages:

- These detectors require a prining step before the interest points selection, which aims to threshold each point for a quality measure. Therefore, the complexity of this approach is increase. However, this procedure reinforce the efficiency and accuracy of the interest points detection [202].
- They only allow for the detection of feature points within a single fixed scale, which leads to robustness detection when the 3D models are corrupted by the noise, and data transformation, i.e. rotation and translation.
- These methods lack an important consideration in human perception: a mechanism take account of global shape information [112]. Some of the methods tend to miss

the the globally-salient points. For some methods, such as [203], the adopted saliency measure can detect keypoints spread quite uniformly over the surface, avoiding only highly planar areas. For HKS method, it only can be applied on meshes with a limited number of vertices due to its high memory requirements.

2.6.2 Scale-invariant Detectors

By contrast with the fixed-scale detector, the common structure of scale-invariant detectors includes building a scale-space defined by some combination of the surface's features [204, 205], and the concept of defining for 2D images is directly extended to the case of 3D data. Novatnack and Nishino [197] constructed a model to analyze the geometric scale variability of a given 3D model in the scale-space of a dense and regular 2D representation of its surface geometry, encoded by the surface normals. A normal map is computed by convolving the vector field with Gaussian kernels of increasing standard deviation. In order to extract the corners and edges at different scales, the authors derive the first- and second-order partial derivatives of the normal map. Finally, the natural scale of each feature is identified and all features are unified into a single set.

Zou et al. [198] presented a geometric scale-space of 3D surface shapes using Ricci flow, and the validation of the feature is measured by the magnitude of the Laplacian of the discrete Gaussian curvature. In this approach, salient geometric features are defined collectively on all the detected scale normalized local patches to form a shape descriptor for surface matching purposes. The salient points are selected by computing the local extrema of the difference of Gaussian function defined over a curved surface in geodesic scale space.

Zaharescu et al. [104] proposed a 3D feature detector: MeshDoG. This detector is able to capture the local geometric and/or photometric properties in an economical fashion. Feature detection is a three-step process: A convolution operation on meshes using a Gaussian kernel is defined to propose a scale-space representation. Then by following the non-maximum-suppression, the feature points are selected as the maxima of the scale space across scales. Finally, only the top 5% of the maximum number of vertices from the extrema of the scale space are taken into consideration.

The MeshSIFT algorithm [206] detected local feature locations as scale space extrema. The MeshSIFT algorithm describes the neighbourhood of every scale space extremum in a feature vector consisting of concatenated histograms of shape indices and slant angles. The feature vectors are reliably matched by comparing the angle in feature space.

The 3D-SIFT [207] technique was proposed by Godil et al. A scale space is constructed by applying 3D Gaussian filters with increasingly large scales to the voxelized model after voxelization. A binary function is denoted on the voxelized model; then each layer of the scale space is represented by its convolution with a 3D Gaussian function. The Difference of Gaussian for each level is computed by subtracting the original model from the scaled model at the corresponding level. By searching the DoG space in both spatial and scale dimensions, the extrema points are detected: the extrema points are declared as interest points if they are located on the surface. Finally, the locations of the interest points are mapped back to the original mesh, and the closest vertices are marked as final interest points.

Sipiran and Bustos [208] proposed an extension to meshes of the Harris corner detection method: 3D-Harris. The algorithm suggests that a neighborhood (rings or adaptive) around a vertex. Next, this neighborhood is used to fit a quadratic patch, which is considered as an image. After applying a Gaussian smoothing, derivatives are calculated, which are used to calculate the Harris response for each vertex.

Scale-invariant detectors have the following advantages and disadvantages: Advantages:

- They allow for detecting keypoints at different scales, and for associating to them a characteristic scale used to define the support for the subsequent description stage [209].
- They achieve quality measurements comparable to those of fixed-scale methods. Moreover, these scale-invariant techniques associated with spatial position and scale, which leads to a more robust result [104, 207, 208].
- The geometry of the mesh is not modified during the creation of the scale-space, and there is no addition pruning stage.

Disadvantages:

- Higher computation complexity than fixed-scale methods. The scale selection at each keypoint is required for the scale-invariant detectors, it may be carried our by mean of non-maxima suppression of this term along the scale. The selection of the keypoints by maximizing the quality measurement spatially and across scales [104, 207, 208].

2.7 Summary

Having divided the task of designing our saliency detection system and its applications into surface smoothing, saliency detection, simplification and interest points detection, the reviews of previous research efforts were addressed. However, there are several research challenges to improve all phases:

First, the ultimate goal of surface smoothing is to produce highly smooth surfaces efficiently for other applications, i.e., rendering, segmentation, or simplification, while still preserving the basic overall shape and important local features of the original model. However, smoothing or removing noise while preserving the features of the shape is hard to coexist. A great deal of mesh smoothing algorithms have been proposed in the literature. The isotropic-based methods are usually fast and simple. The anisotrpic-based methods could preserve more sharp features, and are superior to those using isotropic techniques. However, anisotrpic-based methods need significant computational times, and it is not straightforward to assign appropriate parameters to get good results in the algorithms. To this end, we proposed a fast and effective surface smoothing method by adopting the 2D non-local means filter in Chapter 4.

Second, as the *saliency* is not consolidated in 3D domain, and only few works are presented. It is clear from this review that the saliency detection on 3D surface still faces a number of challenges: high computation complexity, or robustness under different data transformations. In the category of pure computation-based saliency detection methods, Lee et al. [3] introduced the concept of mesh saliency as a measure of regional importance for triangle meshes in computer graphics, which is the most impactive method in this domain. However, in their approach the mesh saliency is expressed by a scale-dependent manner using a center-surrounding operator on the Gaussian-weighted mean curvatures. It is a time-consuming operation to estimate the mesh saliency. Furthermore, almost all of these methods [102, 105, 158, 210], simply select regions where the curvature of a surface patch is different than in its immediate neighborhood. Moreover, the detected salient regions are always hard to be satisfied with the requirements of human vision under specific data transformation, such as rotation. In order to determine how salient is a surface region, consideration of its properties only is not sufficient, the consistency of those properties are within other instances of the same model also need to be considered: therefore, the properties are investigated from locally and globally the give surface would be an useful impact factor for an accurate saliency detection system. For this reason, we proposed a novel saliency detection method based on a human visual perceptual-based feature - RIF - (see Chapter 3), to estimate the salient regions locally and globally in Chapter 5.

Third, most of the surface/mesh simplification algorithms distribute samples on the surface according to the geometric properties of the surface, such as curvature. This approach removes the vertices at a time until the target vertex count is reached. It requires an error metric to quantify which vertices need to be removed, and majority of the error metrics used are geometry-based which measures the spatial error due to simplification. Geometric-based simplification methods are powerful, and simplify small, distant, or otherwise unimportant portions of the scene, reducing the rendering cost while attempting to retain visual fidelity. However, it is hard to quantify the visual fidelity, so most simplification algorithms settle for geometric measures of quality. Fidelity of the simplified surface may be assumed to vary with the distance of that surface from the original mesh. Therefore, sometimes people are also considering the most important measure of fidelity is not geometric but perceptual: does the simplified version look like the original one? In Chapter 6, we present a geometric simplification algorithm guided by perceptual metrics (salient regions).

Fourth, similar to the simplification, most of the existing interest point detection algorithms also rely on the surface geometric properties. In general, the detection of 3D interest points can be treated as extension of the correspondent task performing on 2D images - SIFT [191]. However, it is not straightforward by such extension, and only few works have shown their effectiveness [207, 208]. More recently, the interest point selection is focused by exploiting the notion of saliency in the 3D domain [3]. In Chapter 7, a novel saliency-guided interest point detection method is proposed. This method produces stable and robust results under different data transformation instances.

Chapter 3

A Retinex-based Importance Feature

In this chapter, we will introduce a new feature channel for saliency estimation. Most of the existing saliency detection systems require one or more feature channels from the 2D image or 3D shape to generate the saliency, such as the color, intensity, orientation, curvature, and shape index. In our work, the RIF is proposed. This feature is not based on the surface geometry alone, but also on human visual perception.

3.1 Introduction

Most 3-D images are captured by cameras and scanners. As a result they usually contain imaging noise from a variety of sources: e.g., low contrast, either locally or globally, caused by low, or too strong illumination, or even shadow. Inspired by the principles of visual perception in the field of perceptual psychology and cognitive science, researchers have shown that the visual perception and the comprehensibility of complex 3D models may be greatly enhanced. The benefit of such 3D enhancement is to improve the appearance, remove the noise and de-emphasise unsatisfactory regions of the 3D shape. The Retinex theory will be introduced, and the existing works on Retinex will be reviewed in the next section.

As we mentioned in Chapter 1, our saliency detection model requires two pre-attentive feature channels, which are: the *Retinex-based Importance Feature* (RIF); and the *Relative Distance* (RD). In this chapter, the idea behind the RIF will be introduced.

The RIF is the product of our proposed image enhancement technique. Many previous attempts have been made at producing viable 2D and 3D image enhancement techniques.

FIGURE 3.1: Outline of the process of the RIF.

However, this effort has been complicated by the lack of any general unifying theory of image enhancement, as well as the lack of an effective quantitative standard of image quality to act as a design criterion for an image enhancement system. Furthermore, many enhancement algorithms have external issues, such as problems in fine-tuning the parameters, and excessive dependency on the type of input image. However, judgment of visual quality involves human visual perception, which is quite difficult to model and not yet fully understood.

Figure 3.1 shows the outline of the process of the RIF. The shape index is used as the input data: after the application of the Retinex process a new feature map is produced, which is an enhanced version of the original shape index. Full details will be shown in the following sections.

3.2 Overview of Retinex

We need to be aware that there may be many differences between the image captured by the camera or scanner and the image we actually perceive. For instance, subject to a variety of compensatory factors, we see in terms of the amount of light coming from objects to our eye: the green apple looks green to us during daylight hours, when the main illumination is white sunlight, also at sunset, when the main illumination is red. Perceived brightness depends on factors such as contrast around the pixel, and also on various cognitive processes.

Retinex theory is mainly about the concept of color constancy. Color constancy is an example of subjective constancy, and a feature of the human system of color perception that ensures that the perceived color of objects remains relatively constant under varying illumination conditions. Retinex is precisely such a theory of color constancy.

The word: *Retinex* is combination of the words *retina* and *cortex*. It explains how the visual system extracts reliable information from the world despite changes of illumination. Psychologically, the Retinex theory explains how the colors perceived by human beings are relatively stable, usually irrespective of illumination conditions [18]. In other

words, it is about the human visual perception of external scenes. This theory is defined as a mode of the lightness and color perception of human vision. In other words, it offers a psychological account of the characteristics of human observation of color.

The Retinex theory was originally proposed by Land and McCann in 1971 [211]. It has been influential in the field of the computer vision area. This theory had been adapted and modified for many applications: all of these implementations demonstrate that it effectually improved the perceived quality of the original images, in a manner more consistent with human visual perception.

The Retinex process works initially by decomposing a given image into a reflectance image and an illumination image. The effect is both to produce more visual pleasing results than the original version, and to enhance further image analysis and understanding.

3.3 A computational model of Retinex

The initial idea of the Retinex approach in the field of computer vision was investigated by Land and McCann [211] as a model of the lightness and color perception of human vision. The model was developed starting from the assumption that actual color sensations are related to the intrinsic reflectance of objects, rather than to the radiance values captured by the eyes. A model was then developed, based on a series of experiments with a flat surface composed of color patches and three controllable independent light sources. Through the years, Land evolved the concept from a random walk computation to its final form as a center-surround spatial opponent operation, which is related to the neurophysiological functions of individual neurons in the primate retina, lateral geniculate nucleus, and cerebral cortex [212].

The computational model of Retinex aims to recovery of color constancy. It decomposes a given image into a reflectance image and an illumination image:

$$I(x, y) = L(x, y) * R(x, y) \tag{3.1}$$

The image $I(x, y)$ is the product of two components: illumination $L(x, y)$ and reflectance $R(x, y)$. Generally, $L(x, y)$ is determined by the illumination source and $R(x, y)$ is determined by the characteristics of the imaged object.

The reflectance image contains information about the object: it reveals the object of interest more objectively and can thus be regarded as the enhanced image. To estimate the illumination image, most of the existing studies [213, 214, 215] used the smoothing method, in which the weights of pixels are determined as functions of gradients and

inhomogeneities [213]. Based on the assumption that the illumination varies slowly across different regions of the image, and that the local reflectance may change rapidly across different regions, the processed illumination should be drastically reduced by high-pass filtering, while the reflectance after this filtering should still be very close to the original reflectance. The reflectance may also be found by dividing the image by the high-pass version of the original image, which represents the illumination components.

The benefits of such decomposition include the ability to remove the illumination effects: to enhance the image, which includes spatially varying illumination; and to correct the colors in the image by removing illumination-induced color shifts. Many applications have adapted the Retinex algorithm: examples include image editing, multi-spectral image fusion and high dynamic range compression. In application, the result is an improvement in the visibility of dark object areas, while the visual differentiation of the light areas is maintained.

3.3.1 Previous work

Following the initial work, many versions of Retinex were developed. Their common principle is to assign a new value to each pixel in the image based on spatial comparisons of light intensities. They differ in the order in which the pixels are addressed, as well as in their choice of distance weighting functions. These Retinex-based algorithms may be classified into three groups: path-based algorithm, Poisson equation-based algorithm, and surround-based algorithm. In the first version, path-based, the new pixel values depend on the computation of ratios and products along paths in the image. In the Poisson equation-based algorithms, the new pixel values depend on the thresholding function. The final group includes the center-surround versions of Retinex, in which the new pixel value depends on the ratios of the pixels that are included in the surrounding area.

3.3.1.1 Path-based algorithms

The first group is that of the path-based algorithms, first proposed by Land [211, 212, 216]. Land's algorithm computed subsequent additions of pixel differences along a set of one-dimensional random paths contained in the image. The new value of each pixel was computed as the average over all paths.

Horn [18] reformulated Land's Retinex theory and showed that the illumination can be estimated using a two-dimensional Laplacian. Hurlbert [217] not only formalized the lightness problem, such as retrieving the perceived sensation of color from absolute

intensities, but also showed that most of the algorithms that aimed to solve it could be expressed with a single equation. Brainard and Wandell [218] described the path version formally, using stochastic methods. They studied the convergence properties of Land's Retinex and showed that, as the number of paths and their lengths increases, the result converges to a simple normalization.

The main drawbacks of path-based Retinex methods are their high computational complexity, and the large number of free parameters, such as the number of paths, their trajectories and their lengths.

3.3.1.2 Poisson equation-based algorithms

In Poisson equation-based formulations [216, 219], threshold functions are usually utilized to eliminate the illumination. The reflectance may then be recovered by solving Poisson equations, which can be done by effective algorithms such as the Laplacian or FFT.

The poisson equation can be defined as:

$$\Delta\widehat{l} = \tau\left(\Delta s\right) \tag{3.2}$$

where $\tau\left(\Delta s\right)$ is the clipping operation. Horn [18] suggested an iterative procedure as the solution of the resulting Poisson equation. This method effectively inverts the Laplacian operator. Blake [124] proposed a method to extract the discontinuities from the image gradient magnitude instead of the Laplacian, and thereby developed better boundary conditions that deal with less trivial scenarios along the image boundary. The same equation is actually called ' Poisson-equation-type Retinex algorithm ' in [216], which refers to Kimmel et al. [219]'s variational model for Retinex. This last model is similar to the Horn model. It also assumes that the illumination field is smooth, and inserts into the variational model, a knowledge of the limited reflectance dynamic range as a constraint.

These authors also presented a fast multi-resolution solution to the variational problem:

$$\tau\Delta s = \begin{cases} \Delta s & \text{if } where\, |\Delta S| < T \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

However, extra nonsparse divergence-free vector fields are introduced to the gradient when we solve the Poisson equations, and thus the recovered reflectance is usually not piecewise constant as expected.

3.3.1.3 Surround-based algorithms

Surround-based Retinex computational models are non-iterative. Each pixel is selected sequentially and treated only once. Due to the lesser degree of computational complexity than we find in the other two versions, this version is widely used in the image enhancement area. We will introduce here the basic methodologies of this approach. The implementation results will also be shown.

- Jobson et al. [220] proposed a single scale Retinex (SSR) that employs a simple linear filter with Gaussian kernel to estimate the neighborhood illumination. The SSR computes the new value of each pixel by taking the ratio between the treated pixel and a weighted average of its neighbors, whose weights are given by a Gaussian function. In addition, a logarithmic transformation is employed to compress the dynamic range, the reflectance component R is estimated as the logarithm of the ratio of I and F. The reflectance image takes the following form:

$$R_{SSR}(x,y) = logI(x,y) - log[F(x,y) * I(x,y)] \tag{3.4}$$

  where R_{SSR} is the Retinex output, $*$ denotes the convolution operation and $F(x,y)$ is the surround Gaussian function

$$F(x,y) = Ke^{-r^2/c^2} \tag{3.5}$$

  in which c is the Gaussian surround space constant, and K is determined such that $\int\int F(x,y)dxdy = 1$.

- However, strong shadows cast from a direct light source violate the assumption that the illumination varies only slowly, and halo effects are often visible at large illumination discontinuities in I. In order to solve this problem, Jobson also extended SSR to multi-scale Retinex (MSR) [221] by combining several low-pass filtered copies of the logarithm of the Retinex image, using different cut-off frequencies for each low-pass filter. MSR improved on the previous method by estimating illumination as a combination of several weighted (w_n) Gaussian filters with $n-$th scales (N). The reflectance image is defined by:

$$R_{MSR}(x,y) = \sum_{n=1}^{N} w_n\{logI(x,y) - log[F(x,y) * I(x,y)]\} \tag{3.6}$$

- Rahman et al. [222] in turn extended the MSR with color restoration (MSRCR). The multiscale version is an extension of the SSR that aims to reduce halo artifacts

induced by the single-scale method. It is obtained by averaging three SSRs, using three different spatial constants. MSRCR is given by:

$$R_{MSRCR}(x,y) = \alpha_i(x,y)\sum_{n=1}^{N} w_n\{logI(x,y) - log[F(x,y) * I(x,y)]\} \tag{3.7}$$

where $\alpha_i(x,y)$ is the color restoration factor coefficient in the i_{th} spectral band, which is based on the following transform:

$$\alpha_i(x,y) = f(I_i(x,y)\sum_{n=1}^{N} I_n(x,y)) \tag{3.8}$$

where N is the number of spectral bands, and $f()$ is some mapping function. The color restoration factor is introduced to compensate for the loss of color saturation inherent in this method.

- In addition, Meylan [223, 224] proposed a novel method by including all spatial constants into a single filter to replace the three single-scale Retinex used by Rahman. This approach is applied to the luminance channel, and no color restoration is required. Chen et al. [77] were also inspired by the MSR method to improve the luminance and chrominance contrast of the image, while avoiding dramatic white balance changes and artifacts. This method adapts the technique of global tone mapping, using a circular curve combined with gamma correction, luminance and chrominance contrast enhancement using a modified MSR.

3.3.2 Discussion of Retinex for 2D image applications

The discussion in this section suggests that the previous Retinex methods are actually very similar. They mainly consist of two steps: estimation of the reflectance; and normalization of the illumination. They are all based on the spatial smoothness assumption of the illumination. All the above algorithms apply various, potentially nonlinear, smoothing operators to input image. As mentioned in the previous part, illumination L is estimated as a smooth version of input I. Smoothing should especially be carried out among pixels which have homogeneous illumination, because of illumination discontinuities. This robustness requirement implies that the estimated illumination must be discontinuous at locations where the input image I has strong discontinuities of intensity.

Once the estimation is completed, illumination is normalized by taking the difference between the logarithms of the input image and the estimated illumination. The logarithmic function converts multiplicative noise to additive noise, and makes it possible to promote the useful signals. The estimation of illumination is the core procedure.

Eventually, the reflectance image is yielded by skinning the illumination from the given image, which is then expected to be free of non-uniform illumination, and be a more pleasing image.

All of these methods provided convincing results in dealing with grey or color images. The influence of a bright area on a neighboring dim area was decreased, thus preventing halo artifacts. Using Retinex methods, as shown in Figure 3.2, even images with strong cast shadow are effectively normalized by three different classic Retinex methods.

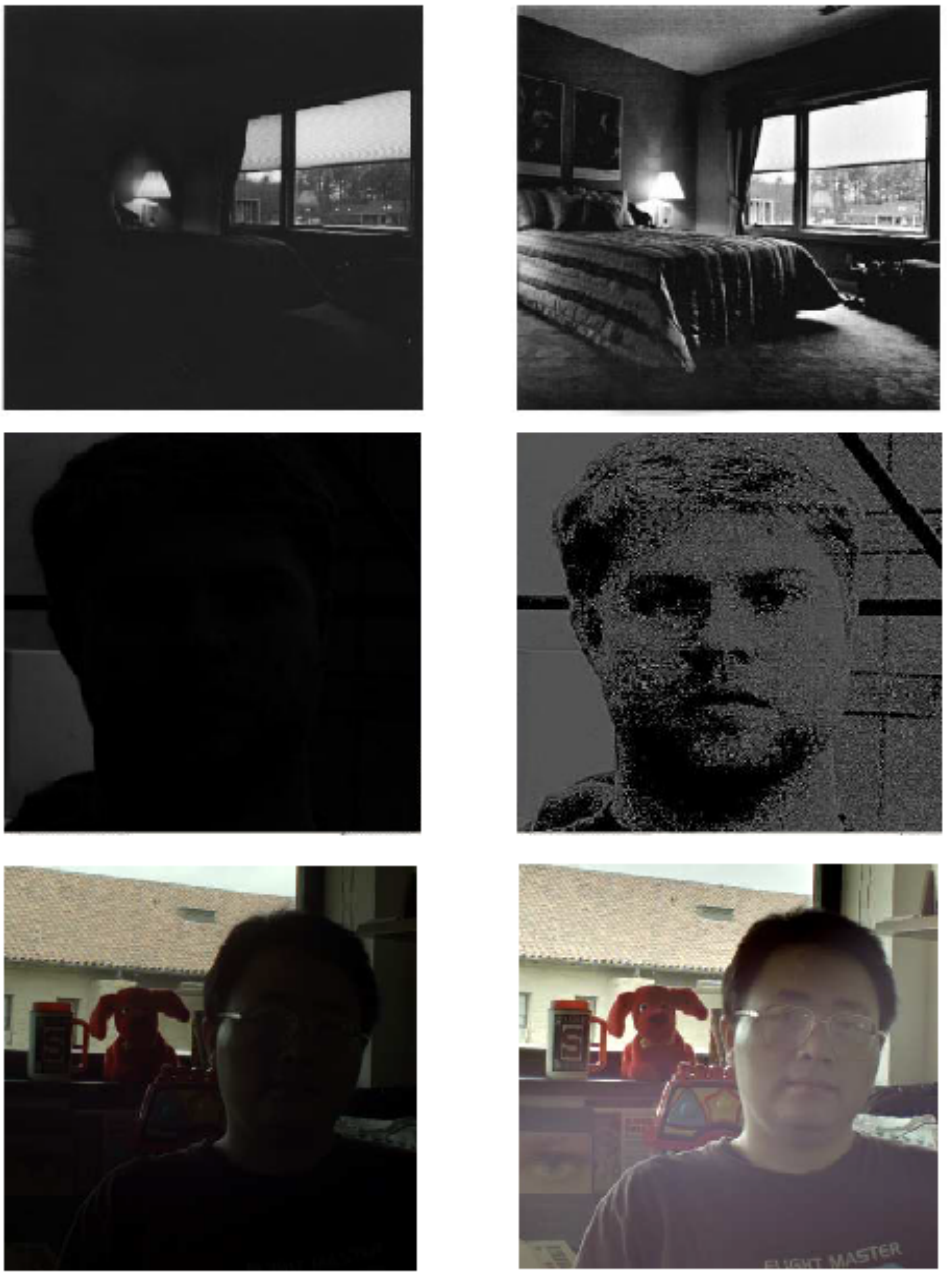


FIGURE 3.2: Examples of different image enhancement methods based on the Retinex. Top: [221]. Middle: [222]. Bottom: [77]. Left: original images; Right: enhanced images.

3.4 Retinex in 3D applications

By better understanding how the brain estimates and represents 3D shapes, we can use the knowledge of Retinex theory to guide many applications in graphics, from representing geometry, to rendering, to improving digital visualizations of 3D objects. By knowing what the brain cares about - what it gets right and wrong - we may be able to predict and therefore compensate for errors in human 3D shape estimation. So far, we know that the optics of the eye project the 3D world onto a 2D image plane on the retina.

Retinex theory deals with the removal of unfavorable illumination effects from images in order to improve their quality. However, the Retinex theory as applied to 2D images focuses on color and illumination only. When it is extended to 3D shapes, it might be employed to enhance shape information: global shape and local geometrical details. Through a large number of experiments from our previous work [225], it has been found that the Retinex theory can also be used for 3D shape analysis, especially for geometrical information enhancement.

Normally, human perception and objective information with respect to vision are not in agreement. The human brain interprets an image of a given 3D shape differently from how photo-sensors or scanners may sense them, by automatically correcting brightness, and removing noise, shadows, glare, or reflections. After the application of Retinex on the shape index map (details are shown in Section 3.4.2), i.e., RIF, component of surface can be used to represent more faithfully the shape index.

This chapter proposes a bilateral filtering-based Retinex process for shape index map enhancement. Figure 3.3 shows the outline of our Retinex computation model to compute the RIF. It is also in line with most of the Retinex computation models used in 2D imaging [226]. The only difference between the traditional algorithm and our own is that the shape index map, rather than the original model is used as the input, because in our project a Retinex-based importance feature has to be estimated, and the Retinex process is applied with a view to enhancing geometric features, rather than the surface appearance.

In this framework, the estimation of reflections is the core procedure, so the proper smoothing method must be carefully selected. In our project, the bilateral filter is used for the computation of the reflectance. This result stands as a theoretic justification for the recently proposed heuristic use of the bilateral filter for Retinex in Elad's work [215].

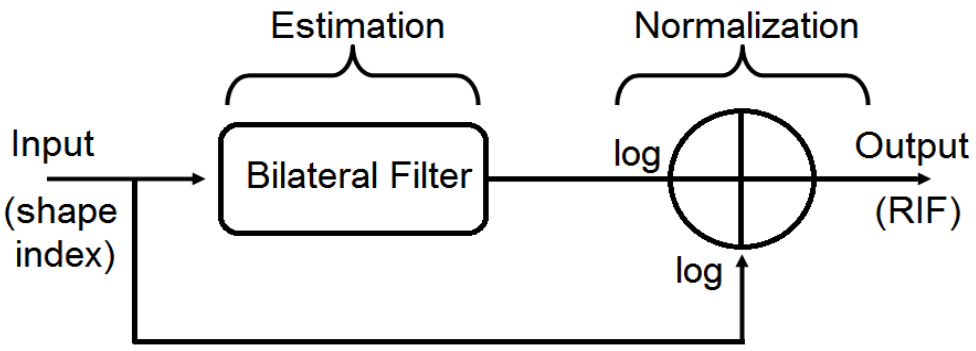


FIGURE 3.3: The outline of the proposed Retinex-based importance feature estimation.

As we mentioned above, the shape index map is used as the input for the Retinex computation model. Therefore, the curvature and shape index estimation are provided in the following sections.

3.4.1 Curvature

Curvature, defined in 3D space, is the measure of how much the curve 'bends' at a single point, and the curvature of a surface intrinsically describes the local shape of that surface. This can be thought of as the rate of change of the angle formed between the tangent and the curve as the tangent is drawn along the curve. The definition of curvature has been modified throughout history; it changes minutely depending upon how many dimensions are being observed, as well as on what specific curve is involved. Curvature is well defined for continuously differentiable lines and surfaces and a number of different curvature measures are defined in differential geometry. Indeed, the curvature variations strongly influence the intensity image derived from the rendering of the object.

Curvature estimation methods generally fall into one of two main categories. The first category, surface fitting [227, 228], involves finding a parametric surface patch fitted to the neighborhood of each data point. The second category, the discrete method [164, 229], involves developing discrete approximation formulas based on the definition of curvature in order to operate on the triangulated data directly. In practice, due to the high computation costs of surface fitting, the discrete method seems more popular in curvature estimation.

Curvature is a very important property of 3D meshes, and there are already a number of curvature estimation approaches that generalize differential-geometry-based definitions of curvatures to discrete meshes [123, 229].

The main types of curvature are mean curvature and Gaussian curvature. Mean curvature is the most relevant to applications at this time and is, as a result, the most studied. Gaussian curvature is regarded as an intrinsic property of space that is independent of the coordinate system that is used to describe that space.

If there exists a surface in three-dimensional space, then at any given specific point there is a plane tangent to that surface. A generalization of curvature known as normal section curvature can be computed for all directions of that tangent plane. By calculating all the directions, a maximum and a minimum curvature value are obtained. Mean curvature and Gaussian curvature are the products of maximum and minimum value, respectively. In this dissertation, Meyer's work is employed to generate the curvature as a surface invariant. In the following chapter, we offer a broad introduction to this work.

3.4.1.1 Meyer's Work

Taubin [123] proposed a very classic curvature generation approach in 1995. In his method, principal curvatures and principal directions are obtained by computing the eigenvalues and eigenvectors of certain 3-by-3 symmetric matrices defined by integral formulas in closed form, and closely related to the matrix representation of the tensor of the curvature. The principal curvature can be obtained as functions of the nonzero eigenvalues of symmetric matrices.

Due to the irregular tessellation of the given mesh and the fact that their range of values may depend on the sampling density, methods to compute curvature based on Taubin's work may produce unstable results. The main reason for these shortcomings is that the curvature tensors are usually computed on the one-ring neighborhoods of the vertices. Meyer [229] proposed a method to compute the mean curvature accurately by employing the Laplace-Beltrami operator on the 3D meshes. In our project, this curvature estimation is employed. In the following section, the rough idea of Meyer's work will be illustrated.

The Laplace-Beltrami operator $L(p)$ is defined as:

$$L(p) = 2k_H(p)n(p) \tag{3.9}$$

where k_H denotes the mean curvature of vertex p in the mesh, $n(p)$ denotes the normal vector. If $k_H(p) = 0$, it shows that the surface area is minimized. The direct relation between surface area minimization and mean curvature is:

$$2k_H(p)n(p) = \lim_{diam(A)\to 0} \frac{\nabla A}{A} \tag{3.10}$$

where A is an infinitesimal area around a point p on the surface, ∇ is the gradient operator with respect to coordinates of p, and$diam(A)$ is the neighborhood diameter. For any vertex p , we select all the triangles in its 1-ring neighborhood and connect the center of each triangle to the midpoints of edges adjacent to p. The infinitesimal neighborhood area of p is computed. Let α_{ij} and β_{ij} be the angles opposite to the edge in the two triangles sharing the edge x_i, x_j. Summing these areas for the whole 1-ring neighborhood, we can write the total area A_{mixed},

$$A_{mixed} = \frac{1}{8} \sum_{x_j \in N(x_i)} (\cot \alpha_{ij} + \cot \beta_{ij}) \|x_i - x_j\|^2 \tag{3.11}$$

Now that the mixed area is defined, we can express the mean curvature normal operator L defined in Equation 3.9 using the following expression:

$$\mathrm{L}(x_i) = \frac{1}{2A_{mixed}} \sum_{x_j \in N(x_i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (x_i - x_j) \tag{3.12}$$

We can easily compute the mean curvature value k_H from the expression above by taking half of the magnitude of this last expression:

$$k_H(x_i) = \frac{1}{4A_{mixed}} \sum_{x_j \in N(x_i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (x_i - x_j) \tag{3.13}$$

In addition, the Gaussian curvature k_G can be estimated:

$$k_G(x_i) = \frac{(2\pi - \sum_{j=1}^{\#f} \theta_j)}{A_{mixed}} \tag{3.14}$$

where θ_j is the angle of the j-th face at the vertex x_i. $\#f$ denotes the number of faces around this vertex.

3.4.2 Shape Index

The local property of an analytical shape is most easily accessed through its local differential structure. In the following we briefly introduce a shape invariant: shape index. Koenderink [230] has suggested that the shape index might offer a simple measurement of the local shape. It travels from a concave sphere through a concave cylinder, a balanced saddle, and a convex cylinder to a convex sphere as the index increases. This feature is useful for representing an object based on its constituent shape forms. Unlike

the mean and the Gaussian curvature, the shape index is invariant to scaling of the shape. The most appealing property of shape index is that it is scale, translation, and rotation invariant. Moreover, shape index strongly emphasises points where surfaces deviate from being smooth, even for small changes. Therefore, the representing surface with its shape index values offers more clearly visible shape details. The shape index θ may be calculated as:

$$\theta = \frac{2}{\pi} \arctan \frac{k_2 + k_1}{k_2 - k_1} \tag{3.15}$$

where k_1 and k_2 are the minimum and maximum principal curvatures at point P. Therefore, since both mean curvature and Gaussian curvature have been derived for triangulated surfaces in the last section, we may define the discrete principal curvatures as:

$$k_1(x_i) = k_H(x_i) + \sqrt{\Delta(x_i)} \tag{3.16}$$

$$k_2(x_i) = k_H(x_i) - \sqrt{\Delta(x_i)} \tag{3.17}$$

where $\Delta(x_i) = k_H^2(x_i) - k_G(x_i)$. However, in order to avoid any numerical problems, we must make sure that the Δ is always positive, and threshold Δ to zero if it is less than 0.

Figure 3.4 and Figure 3.5 show the three different estimated invariants: mean curvature, Gaussian curvature and shape index, respectively. It may be seen that the shape index has greater ability than the curvatures to represent some of the 3D shapes, both of flatter regions and of more complicated regions, such as the bumpy region of *lobster* and curly hair of *buddha*.

FIGURE 3.4: Examples of the three surface invariants. Left: mean curvature. Middle: Gaussian curvature. Right: Shape index. Models, from top to bottom: *lobster*, *teletubby*, *buddha*, and *bird*. The warm color of mean and Gaussian curvature cases indicates convex areas, and cool color indicates concave regions. The warm color in the shape index case marks the flatter regions, and cool color indicates the high curvature areas.

.

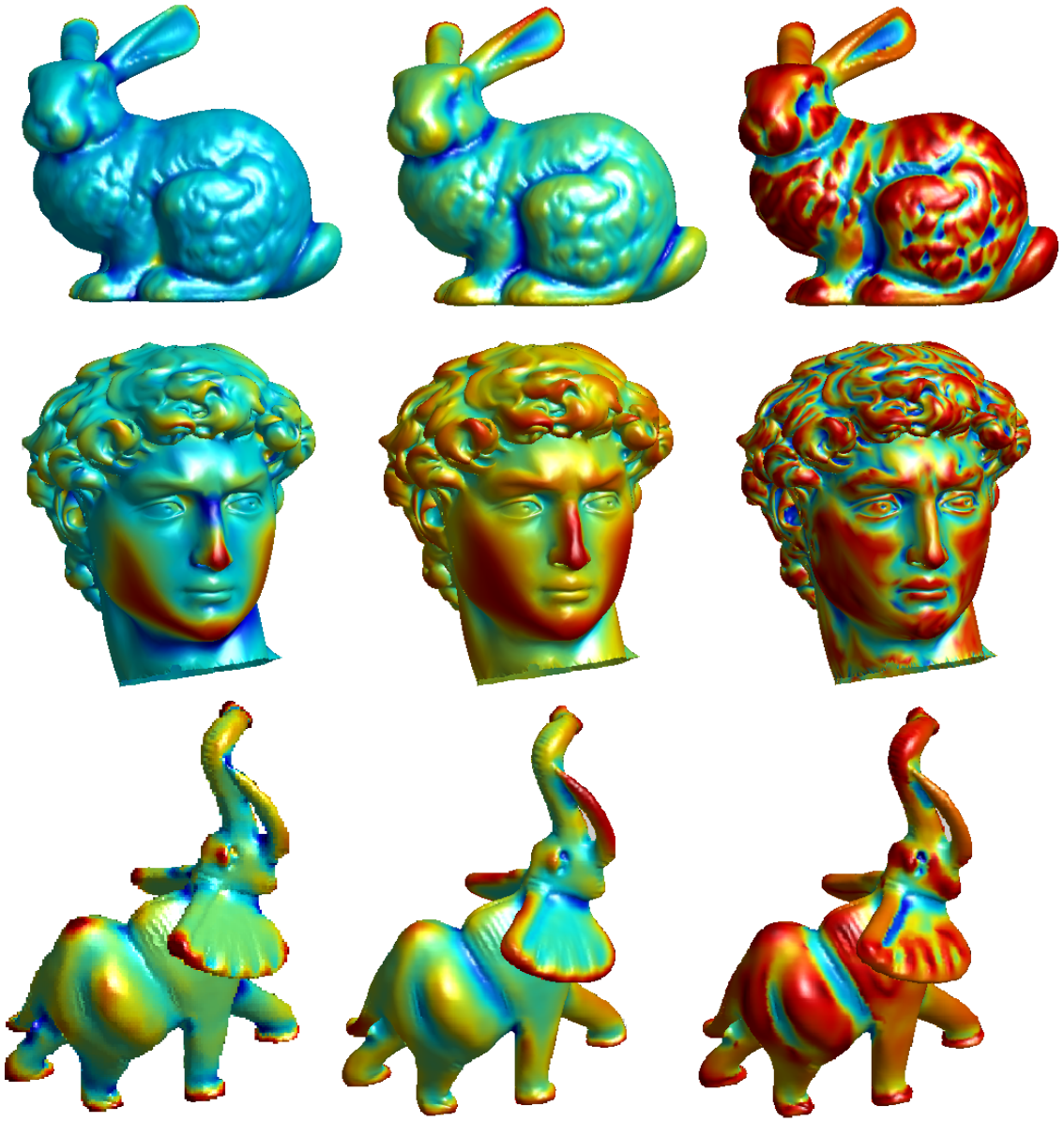


FIGURE 3.5: Examples of the three surface invariants in a different dataset. Left: mean curvature. Middle: Gaussian curvature. Right: Shape index. Models, from top to bottom: *bunny*, *david-head*, and *elephant*. The warm color of mean and Gaussian curvature cases indicates convex areas, and cool color indicates concave regions. The warm color in the shape index case marks the flatter regions, and cool color indicates the high curvature areas.

.

3.4.3 The Retinex-based Importance Feature

The Retinex-based Importance Feature (RIF) is obtained by means of a relative measurement, which is likely to reflect more faithfully the original geometry and to be more resistant to imaging noise. Our work employs a Retinex theory based on a bilateral filter to enhance and optimize the shape index, as the bilateral filter considers not only the spatial distance but also the scalar function difference. Moreover, the bilateral filter has been extended to filter the meshes due to its nonlinear, feature-preserving characteristics [121, 231]. As mentioned above, the bilateral filter also stands as a theoretic justification for the recently proposed heuristic use of the bilateral filter for Retinex in Elad's work [215].

Let K be the shape index values for the given 3D data: and let u be one of the vertices. By taking the difference between the logarithms of the input and the bilateral-filtered shape index, the output of the RIF ($\Re$) is calculated as follows:

$$\Re(u) = log(K(u) + 1) - log(\mathscr{L}(u) + 1) \tag{3.18}$$

where the $\mathscr{L}$ is the bilateral filtering on the shape index.

Tomasi and Manduchi [232] initially defined the bilateral filter for image processing. The definition of the bilateral filter h on a 2D image is:

$$h(x) = \mathscr{N}^{-1}(x)\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\xi)c(\xi, x)s(f(\xi), f(x))d\xi \tag{3.19}$$

where $\mathscr{N}$ is the normalization factor, f is the input image, s denotes the similarity function, it operates in the input image f, ξ and x indicate the coordinates of two pixels, and $c(\xi, x)$ measures the geometric closeness between neighborhood center x and a nearby point ξ.

It is worth noticing that there are two integrations $\int$ in the equation, because in this case, the filter is dealing with 2-dimensional images. Thus, when the bilateral filtering is adopted to deal with a shape index, only one integration is required, as the dimension of the shape index is one: each vertex has one shape index value. Hence, the bilateral filter in our case may be rewritten as:

$$\mathscr{L}(u) = \mathscr{N}^{-1}(u)\int_{-\infty}^{\infty} K(\ell)g(\ell, u)s(K(\ell), K(u))d\ell \tag{3.20}$$

with the normalization $\mathscr{N}$

$$\mathscr{N}(u) = \int_{-\infty}^{\infty} g(\ell, u)s(K(\ell), K(u))d\ell \tag{3.21}$$

where $g(\ell, u)$ measures the geometric closeness between a vertex u and a nearby vertex ℓ, and function s measures the similarity between u and ℓ. Shift-invariant Gaussian filtering is employed to measure the similarity in the origin work of the bilateral filter [232], and it also has been used in our work. Both the geometric measurement g and similarity function s are Gaussian functions of the Euclidean distance between their arguments. More specifically, g and s may be defined as follows, respectively :

$$g(\ell, u) = e^{-\frac{1}{2}(\frac{d(\ell,u)}{\sigma_d})^2} \tag{3.22}$$

$$s(K(\ell), K(u)) = e^{-\frac{1}{2}(\frac{d(K(\ell),K(u))}{\sigma_r})^2} \tag{3.23}$$

where σ_d shows the spatial spread based on the desired amount of low-pass filtering, and σ_r is the geometric spread in the image range that is set to achieve the desired amount of combination of shape index values. $d(\ell, u)$ denotes the Euclidean distance between ℓ and u:

$$d(\ell, u) = \|\ell - u\|, \tag{3.24}$$

$d(K(\ell), K(u))$ measures the distance between two shape index values $K(\ell)$ and $K(x)$:

$$d(K(\ell), K(u)) = \|K(\ell) - K(u)\|, \tag{3.25}$$

this could simply be the absolute value of the shape index difference.

The bilateral filtering replaces the shape index value at vertex u with an average of similar and nearby shape index values. In the smooth regions, shape index values in a small neighborhood are similar to each other, and the normalized similarity is close to one. Therefore, the bilateral filtering averages away small, weakly correlated differences of shape index. The normalization term $\mathscr{N}$ ensures that the weights add up to one for all shape index values. As a result, the filter replaces the large shape index at the centre by an average of the large shape index value in its vicinity, and vice versa.

In line with the assumptions of the Retinex theory, the reflectance $\Re$ is restricted to be in the range $\Re \in [0, 1]$. Thus, we normalized $\Re(u)$ into the range $[0, 1]$ by

$$\Re(u) = \frac{\Re(u) - \Re_{min}}{\Re_{max} - \Re_{min}} \tag{3.26}$$

where $\Re_{min}$ and $\Re_{max}$ are the minimal and maximal values of $\Re(u)$.

3.5 Results and Summary

Figure 3.6 and Figure 3.7 illustrate the RIF maps for different 3D models. Overall, the flat, convex, and concave regions of demonstrated results are distinguishable for human observer. It is shown that this method can discriminate clearly between flat, concave and convex regions. Areas such as the bumpy area of *lobster*, the eyes of *teletubby*, the eyes and wing of the *bird*, the nose, mouth and eyes of *buddha* are easily recognised. The bottom two rows of Figure 3.7 illustrate the RIF results on complete models. Similar to partial models, the local details are also more distinguishable. For instance, the hair texture of the *bunny*, the curly hair and mouth of *lion*, the ears, eyes of *Armadillo*, the hair texture and face region of *david*. Figure 3.7 shows the robustness of the RIF under conditions in which the models were rotated around an unknown rotation axis. It may be seen that local details, such as the eyes and mouth of *buddha*, which are discriminated by the RIF, are retained consistently.

A rapid implementation of enhancement of 3D images by introducing a surface invariant - shape index - has been carried out by adapting the Retinex theory from constancy of color perception to the analysis of a 3D surface. The final formalism involves only two variables: the shape index, and a bilaterally-smoothed surface version. The outcomes of this surface enhancement significantly improved the illustration of surface details, and make for a more visually pleasing image when compared to curvature or shape index representations alone.

The main contribution of this chapter is that the Retinex theory has been adopted into 3D shapes. The benefits of such adoption are the local features and global shapes can be demonstrated more significantly, and the results may be helpful for further shape analysis. However, there is no evaluation metric to validate the RIF so far. Future work will focus on the development of evaluation metric on RIF.

FIGURE 3.6: Examples of RIF on partial (top two rows) and complete (bottom two rows) 3D models. It is easy to identify the local detail of the given shapes when the RIF mapped on.

FIGURE 3.7: Examples of the given shapes mapped with RIF, and the shapes subject to a rotation of θ around an unknown rotation axis, follow by a translation. The local distinctive areas are represented significantly.

Chapter 4

Pre-processing: Surface Smoothing

In this chapter, we aim to extend the non-local means [233] concept to mesh smoothing, and the newly developed mesh smoothing method has a number of important advantages over the main state-of-the-art mesh denoising techniques. Non-local means (NL) is an image de-noising process based on non-local averaging of all the pixels in an image. In particular, the amount of weighting for a pixel is based on the degree of similarity between a small patch centered around that pixel and the small patch centered around the pixel being de-noised. The motivation of using non-local means filter on 3D surface, and the algorithm will be illustrated in this chapter.

4.1 Introduction

With geometry scanners becoming more widespread and a corresponding growth in the number and complexity of scanned models, robust and efficient geometry processing becomes increasingly desirable. The quality of reconstructed surface models is of vital importance in many fields such as reliable path planning and obstacle avoidance in robot navigation and physical realism in model based 3D object recognition and computer graphics. However, the acquired data usually contains imaging noise due to low reflection or specular reflection, occlusion and depth discontinuity. Sometimes, a rough surface is generated due to the rapid change of orientations and vertex locations of reconstructed surfaces caused by noise introduced in the process of surface scanning and image registration and integration. Therefore, it is an active area of research to effectively filter noise from the reconstructed surface models, while preserving the desired level of detail. Mesh smoothing, or denoising, is a process dedicated to the removal of

noise with minimal damage caused to the geometric features of the object. The benefits of such surface smoothing are not only improved visual appearance of a 3D object, but also improved data availability to subsequently process, such as matching, surface segmentation, and mesh simplification.

A wide variety of mesh smoothing algorithms have been proposed in recent years. A review of surface smoothing has been provided in Chapter 2.

4.1.1 Proposed Method

Due to the assumptions made in previous methods of regularity in the original surface, some details and fine features are smoothed out, because they behave in all functional respects as noise. By contrast with most previous approaches, we present in this chapter a surface mesh denoising algorithm based on an extended non-local means filter. This method takes advantage of the high degree of redundancy of 3D surfaces. To the best of our knowledge, there have been no previous works on non-local means filtering of mesh. With this filter, a smoother version may be obtained robustly, since it takes into consideration the similarity between the local neighborhood of the pixel being processed and that of the other pixels. However, when extending the non-local means filter to 3D mesh, one main difficulty is how to define the similarity neighborhood. Yoshizawa et al. [234] constructed a local approximation based on radial basis functions to define the similarity kernel. In this work, we address this problem by fitting a local bicubic B-spline surface [235].

The rest of the chapter is organized as follows. In Section 4.2.1, we present a brief overview of the non-local means approach, and then state the problems encountered in Section 4.2.2 when extending this approach to 3D mesh. In Section 4.3.1, we show the background of the B-spline, which is one of the solutions of the problem stated. The final extended non-local means filter on 3D mesh is presented in Section 4.3.2. The experimental results are demonstrated in Section 4.4, which includes a visual comparison to the results obtained with other smoothing methods, and numerical measurements. Section 4.5 summarises our findings.

4.2 Non-local Means Filtering

The local smoothing methods and the frequency domain filters aim at noise reduction and construction of the main geometrical configurations, but not at the preservation of the fine structure, details and texture [236]. Recently, the so-called non-local means filter

concept, a natural and elegant extension of the image bilateral filtering paradigm, was proposed by Buades et al. [233]. The basic idea behind non-local means is as follows: for a given pixel, its new (smoothed) intensity value is computed as a weighted average of the other image pixels, with weights reflecting the similarity between local neighborhoods of the pixel being processed and the other pixels.

4.2.1 Overview in 2D Applications

In this section, we introduce a 2D image denoising method - the non-local means filter - and extend it to the 3D domain. As we know, several methods had been developed before this study was released. Most of the previous studies emphasized that a wide class of studies share the same basic approach: denoising is achieved by averaging, e.g., by means of Gaussian filtering, anisotropic filtering and neighborhood filtering. Natural images often contain many structured patterns, which may be misclassified either as details to be preserved or as noise, when usual neighborhood filters are applied. As a result, given the assumptions made by these methods concerning the regularity of the original image, details and fine structures are smoothed out, because they behave in all functional respects as noise.

The non-local means filter takes advantages of the high degree of redundancy of any natural image. For a given pixel, the restored gray value is obtained by the weighted average of the gray values of all pixels in the image; each weight is proportional to the similarity between the local neighborhood of the pixel being processed and the neighborhood corresponding to the other image pixels. Several accounts of successful non-local means filter-based smoothing have been published [237, 238, 239, 240, 241].

The basic idea underlying this method of 2D image smoothing is that it assumes an extensive amount of self-similarity in an image, then reduces the high frequency surface information, which tends to smooth and flatten the surface. Thus, similarity is measured based on the geometrical configuration in the neighborhood instead of on a single pixel itself.

Let I be a given image. Based on a weighted average of all pixels in its neighborhood Θ, the non-local value $I(x_1)$ at pixel x_1 can be estimated by

$$NL(x_1) = \frac{\sum\limits_{x_2 \in \Theta} \psi(x_1, x_2) I(x_2)}{\sum\limits_{x_2 \in \Theta} \psi(x_1, x_2)} \tag{4.1}$$

where the weight ψ is computed by the similarity of the Gaussian neighborhood between pixels x_1 and x_2:

$$\psi(x_1, x_2) = exp(-\frac{\| \Theta_{x_1} - \Theta_{x_2} \|^2_{2,a}}{h^2}) \tag{4.2}$$

Θ_{x_1} is a square neighborhood centered at pixel x_1, $\| \cdot \|_{2,a}$ is the Gaussian weighted Euclidean distance function, a is the standard deviation of the Gaussian kernel and h is the decay parameter (acts as a degree of filtering). Most of the existing works on non-local means filter use the intensity gray level vectors to compute the similarity. In this paper, the similarity will be obtained from the RIF of the surface. As we mentioned in the Chapter 3, the RIF has the ability to describe the local details, and it is possible to preserve the local detail after a non-local means filter is applied to the RIF.

4.2.2 Problems Arising in 3D Applications

The non-local means filter not only compares the gray level at a single point, but the geometrical configuration in a whole neighborhood. This allows a more robust comparison than is afforded by the use of neighborhood filters. However, when extending the non-local means filter approach to process a 3D mesh, two problems arise.

Determination of $\psi(\cdot, \cdot)$. 2D images usually have a regular structure, which in most cases is not true for a mesh, due to variations of sampling density in the range scanning process. Yoshizawa et al. [234] also employed a non-local means filter to smooth mesh. They used radial basis functions (RBFs) to build a local approximation of the mesh in a neighborhood, and thus to define the similarity kernel. However, their work involves a variety of parameter definitions, which is an inconvenience when seeking to identify an approximate parameter to implement the smoothing process. Therefore, in our work, we employ an alternative approach to define a regular structure in order to obtain the similarity kernel of non-local means on a 3D mesh: the B-Spline surface.

Determination of $I()$. Equation 4.1 shows the result of pixel x_1 after non-local means filtering, in which the calculation is based on the similarity between x_1 and x_2. It may be seen that the similarity is carried out because of the mutation of pixels. However, when we apply the non-local means filter to a 3D surface, there is no variant that can be used directly to illustrate the difference between two vertices. In this study, we use the distance between vertex u and the projection position of vertex v on unit normal at u as I_v.

4.3 Non-local Means Filter in 3D Surface Applications

In our work, the B-spline surface is employed to determine the similarity neighborhood, which generates the control net for the input mesh. The advantage of using B-spline surfaces is that the underlying control net is topologically similar to the image grid-structure. Consequently, the non-local means filter may be applied to the grid structured mesh, and the similarity between neighborhoods in 3D may be more easily measured by computing the corresponding 3D points on the surface when given uniformly parametric u and v in the $u-v$ space of the B-spline surface.

4.3.1 B-spline Surfaces Optimization

Before considering the B-spline surface, we must first introduce the Bezier curve and the B-spline curve. The fundamental idea behind the Bezier curve [242], the B-Spline curve [235], and the B-spline surface [235] involves using control points/polygons to display the origin curve/surface graphically: allowing thus the curve/surface to be manipulated intuitively.

Bezier Curve. The most basic Bezier curve is made up of two end points and control handles attached to each node. The control handles define the shape of the curve on either side of the common node. Bezier curves offer well-behaved control within a convex hull for any degree. The definition of a Bezier curve of degree n is given by

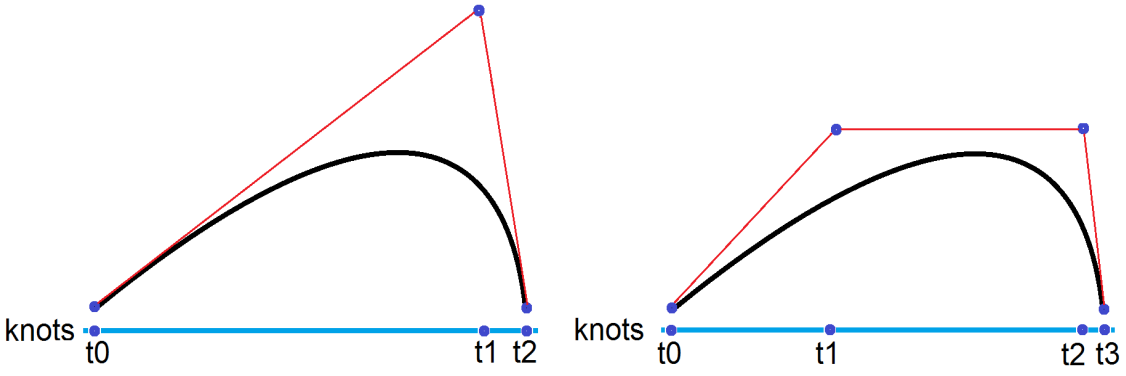
$$C(t) = \sum_{i=0}^{n} P_i B_{i,n}(t) \tag{4.3}$$

where P is the control points set: for example, a cubic Bezier curve consists of four control points used to position and modify the curve. $B_{i,n}(t)$ is the basis function

$$B_{i,n}(t) = \begin{pmatrix} n \\ i \end{pmatrix} t^i (1-t)^{n-i} \tag{4.4}$$

where i is the control point to be weighted, and n is the degree of the curve. The t is termed the *knots* values. The *knot sequence* is a set of non-decreasing real numbers. Figure 4.1 shows Bezier curves with three and four control points, and with their knot sequences t_n.

The B-Spline Curve is similar to the Bezier curve, which is a linear combination of control points. B-spline curves have an advantage over Bezier curves in that they are smoother and easier to control. If any one of control points in a Bezier curve is changed, the whole curve will be affected: but this is not the case for the B-spline. B-splines



(a) (b)

FIGURE 4.1: Examples of the Bezier curve and their knot sequences. (a) Quadratic Bezier curve. (b) Cubic Bezier curve.

consist entirely of smooth curves, but sharp corners can be introduced by joining two spline curve segments. The continuous curve of a b-spline is defined by control points.

In practical terms, B-spline curves may be thought of as a method for defining a sequence of degree n Bezier curves that join automatically, regardless of where the control points are placed. Moreover, a B-spline has the advantage of compact presentation and offers great economies in manipulation and storage.

The B-spline curve is defined as

$$Q(t) = \sum_{i=0}^{n} P_i N_{i,k}(t), \tag{4.5}$$

where P is the control point vector, n is the number of control points, $N_{i,k}(t)$ is the basis functions, and k is the degree of the curve, $t_{k-1} \leq t \leq t_{k+1}$ and $2 \leq k \leq n+1$. The knot t represents the active area of the real number line that defines the B-spline basis. It takes $k+1$ knots, or k intervals to define a basis function. The shape of the basis functions is dependent only on the knot spacing and not on specific knot values, since the basis functions are based on knot differences. Figure 4.2 shows a B-spline curve with 15 control points.

B-Spline Surfaces are an extension of B-spline curves. The most common kind of a B-spline surface is the tensor product surface. The surface basis functions are products of two curve bases. The surface is a weighted sum of surface (two-dimensional) basis functions. The weights are a rectangular array of control points:

$$Q(t,s) = \sum_{i=0}^{n+1} \sum_{j=0}^{m+1} P_{ij} N_{i,k}(t) N_{j,l}(s) \tag{4.6}$$

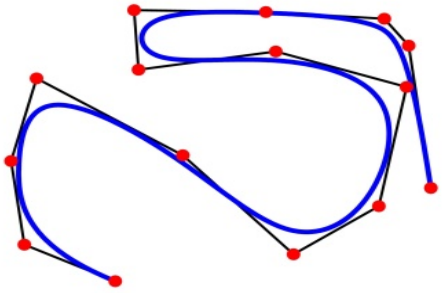


FIGURE 4.2: Example of the B-spline curve with 15 control points.

where

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_i^{k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1}^{k-1}(t) \tag{4.7}$$

$$N_{i,1}(t) = \left\{ \begin{array}{ll} 1, & t_i \leq t \leq u_{i+1} \\ 0, & otherwise \end{array} \right\} \tag{4.8}$$

$$N_{j,l}(s) = \frac{s - s_j}{s_{j+l-1} - s_j} N_j^{l-1}(s) + \frac{s_{j+l} - s}{s_{j+l} - s_{j+1}} N_{j+1}^{l-1}(s) \tag{4.9}$$

$$N_{j,1}(s) = \left\{ \begin{array}{ll} 1, & s_j \leq s \leq s_{j+1} \\ 0, & otherwise \end{array} \right\} \tag{4.10}$$

where P_{ij} is the control points. Changing the control point does not change the B-spline surface. $N_{i,k}(t)$ is the i^{th} basis function of order k as the function of t, $N_{j,l}(s)$ is the j^{th} basis function of order l as the function of s. $N_j^{l-1}(s)$ denotes jth basis function of order $l-1$ as a function of s. $N_{j+1}^{l-1}(s)$ indicates the $(j+1)th$ basis function of order $l-1$ as a function of s.

The B-splines basis functions possess such important properties as non-negativity, local support, partition of unity and linear independence. Similar to the B-spline curve, the B-spline surface is also a network of polynomial pieces. Each piece of a B-spline surface is a two-dimensionally - represented part of the surface or patch, and each surface of a B-spline may be represented uniformly, as it provides a periodic spacing in each direction. The B-spline surface approximates the shape of the control net: if the control polygon of a B-spline surface is transformed, the curve surface itself is transformed by the same affine transformation. Figure 4.3 shows two B-spline surface examples on different types of shapes.

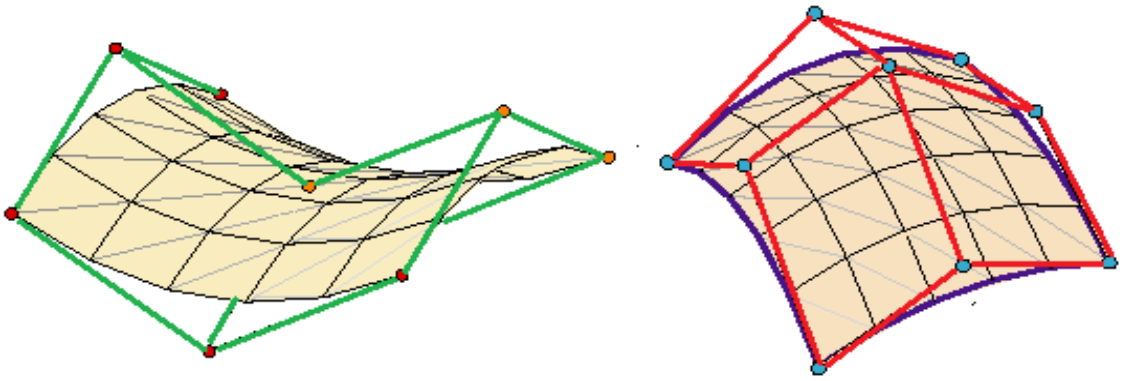


FIGURE 4.3: Example of B-spline surface on saddle and convex shapes.

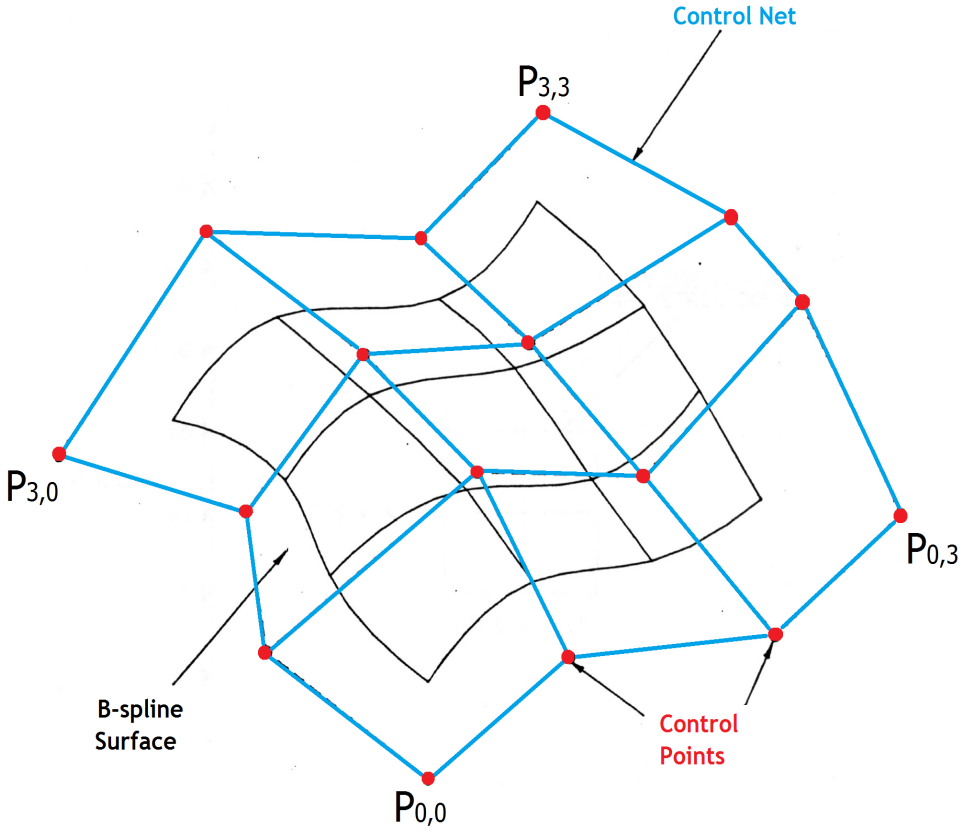


FIGURE 4.4: Bicubic B-spline surface.

4.3.2 Non-local Means Filtering

The 2-ring vertices of v_i are used as input data points and are parameterized by projecting them onto a plane (e.g., the tangent plane of v_i) and scaling them to range $[0, 1]$. The similarity neighborhood of a given v_i is obtained by fitting a local bicubic B-spline surface. Bicubic B-spline is always used in geometric modelling, because of its well-known advantages: result basically from the local support of the basis spline, and

the control points are independent on the knot vector [243]. Figure 4.4 shows a bicubic B-spline surface and the corresponding control net. As determined in [244], $n = 3$ for a bicubic B-spline surface. Thus, the B-spline surface may be rewritten as [245]:

$$Q(t,s) = \sum_{i=0}^{3}\sum_{j=0}^{3} P_{ij} N_{i,k}(t) N_{j,l}(s) \tag{4.11}$$

where P_{ij} are the control points, as shown in Figure 4.4. $N_{i,k}(t)$ and $N_{j,l}(s)$ are the B-spline function in the biparametric t and s directions (see Equation 4.7-Equation 4.10). The non-local means filter may then be applied to the 3D surface, the similarity neighorhood of a given vertex being determined by fitting the bicubic B-spline surfaces. So the similarity between two vertices u and v can be rewritten as:

$$\psi(u,v) = exp(-\frac{\parallel Q_u - Q_v \parallel^2_{2,a}}{h^2}) \tag{4.12}$$

Noise in a 2D image is always indicated by the changing of the pixel values, while noise in the 3D domain is usually revealed by the position difference of the vertices. In consequence, if a non-local means filter is applied to the mesh, a new $I(v)$ is estimated:

$$I(v) =< \mathbf{n}_u, v - u > \tag{4.13}$$

where $v - u$ indicates the Euclidean distance between these two vertices, and $\mathbf{n}_u$ is the unit normal at u. Equation 4.13 shows the difference of the positions between u and v under projection of $\mathbf{n}_u$, this may also be seen in Figure 4.5, in which v' indicates the projection of v on $\mathbf{n}_u$. Finally, mesh filtering with non-local means may be reorganised as

$$NL(u) = \frac{\sum\limits_{v \in Q} \psi(u,v) I(v)}{\sum\limits_{v \in Q} \psi(u,v)} \tag{4.14}$$

We give a heuristic rule for choice of parameters of the delay parameter. In general, for models with higher noise level, the vertices deviate further from their true positions, so h should be larger. Our experiments show that h = 10 is generally enough for models with Gaussian noise with 0.1 mean edge length of standard deviation, and h = 20 enough for 0.2 and 0.3 mean edge length of standard deviation.

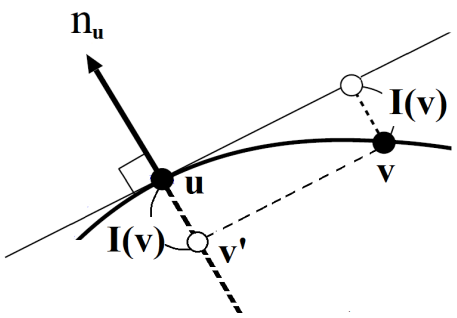


FIGURE 4.5: Position difference of u and v under projection on $\mathbf{n}_u$.

4.4 Experimental Results

This section demonstrates the results of tests carried out on our smoothing approach and comparative studies. We provide visual comparison and several numerical measures by which to compare our approach with other state-of-the-art 3D surface smoothing methods: bilateral filtering [119]; median filtering [246]; the fuzzy vector median filtering [247]. These methods are easily to be re-implemented, effectively, and highly cited. It is worth noting that another non-local means filter based mesh smoothing by Yoshizawa et al. [234] will not appear in our comparison since it is time consuming. They used radial basis functions (RBFs) to build a local approximation of the mesh in a neighborhood, and thus to define the similarity kernel. Their work involves a variety of parameter definitions, which is an inconvenience when seeking to identify an approximate parameter to implement the smoothing process. More comparison of evaluation results between our method and [234]'s will be reported in further publication.

4.4.1 Visual Comparison

First, we present the results of our method when applied to different models, as shown in Figure 4.6. It may be seen that the outcome of the proposed method is more visually pleasing: the flatter regions become smoother, and the local features are well preserved, which is confirmed by the results of applying Retinex, as shown in Figure 4.6(d).

We then make a visual comparison between results obtained using our algorithm with those from several other algorithms: bilateral filtering [119]; median filtering [246]; the fuzzy vector median filtering [247]. All the models were noise-added: Gaussian noise, standard deviation=0.2.

Figure 4.7 shows the denoising results from a complicated human face model *buddha*, which has both flat and curved surfaces. Since this model is a partial surface, which also has sharp edges, it may be seen that the Laplacian smoothing successfully smoothes

FIGURE 4.6: Examples of the proposed non-local means filter applied to different models. Top to bottom: *lobster, buddha, bird, face-rick.* (a) Original surfaces. (b) Retinex surfaces. (c) Smoothed surfaces. (d) Retinex on smoothed surfaces

the flatter regions, such as the cheek of the *buddha*. However, it also smoothes the curly hair region, whereas this region is an important distinguishing characteristic of *buddha*. Moreover, it creates noise elements in the boundary of this model, which make the smoothed model differ significantly from the original. Fuzzy vector median filtering, and our approach both preserve sharp edges and the surface characteristics. There is little visible difference between the latter two results, although our method again seems to produce a smoother surface without blurring details.

Figure 4.8 shows the denoising results from a simple model *bottle*, which, unlike *buddha*, is defined by smooth flat and curved surfaces and edges only. Laplacian smoothing

again introduces spurious additional vertices in the boundary area. Fuzzy vector median filtering, and our approach produce smooth surfaces and preserve most sharp edges, but blur those sharp edges with small angles.

Figure 4.9 shows denoising results for the $fandisk$ model. All four approaches preserve most of the sharp edges. Bilateral filtering tends to blur sharp edges. Median filtering preserves sharp edges, but makes flat areas appear uneven. In contrast, fuzzy vector median filtering and our approach both preserve sharp edges and flat areas. Nevertheless, close examination shows that our approach generates a smoother final surface for this model.

Figure 4.10 shows the denoising results on the *david* model. All approaches do well apart from median filtering. Median filtering has a tendency to enhance features in the noisy model, and the resulting surface is not smooth. Fuzzy vector median filtering, and our approach seem to produce smoother final surfaces than bilateral filtering and median filtering.

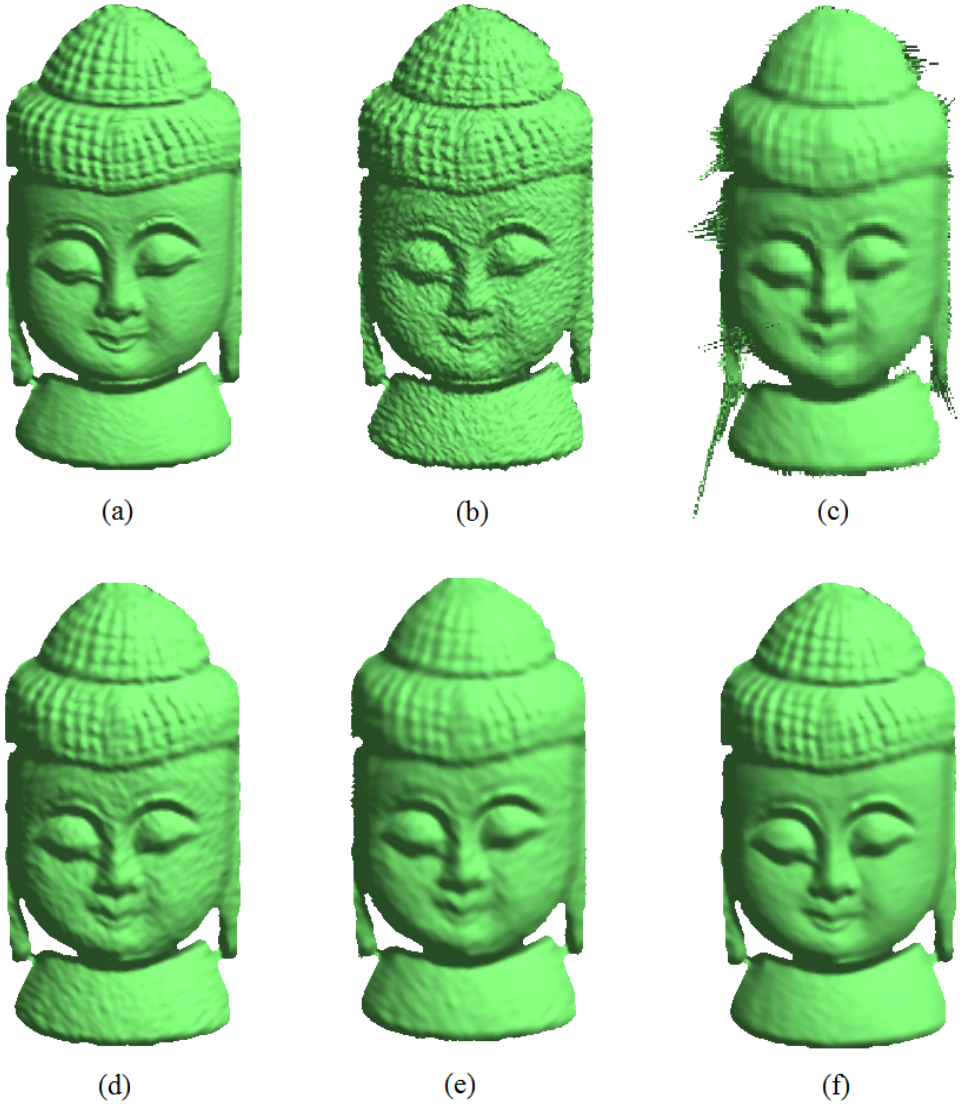


FIGURE 4.7: Smoothing of a noise-added model *buddha*. (a) Original model. (b) Noisy model, with added Gaussian noise, standard deviation=0.2. (c) Bilateral filter results (Iteration times=5). (d) Median filtering results (Iteration times=20). (e) Fuzzy vector median filtering results (First iteration times=20, second iteration times=20, $\sigma = 0.3$). (f) Our result.

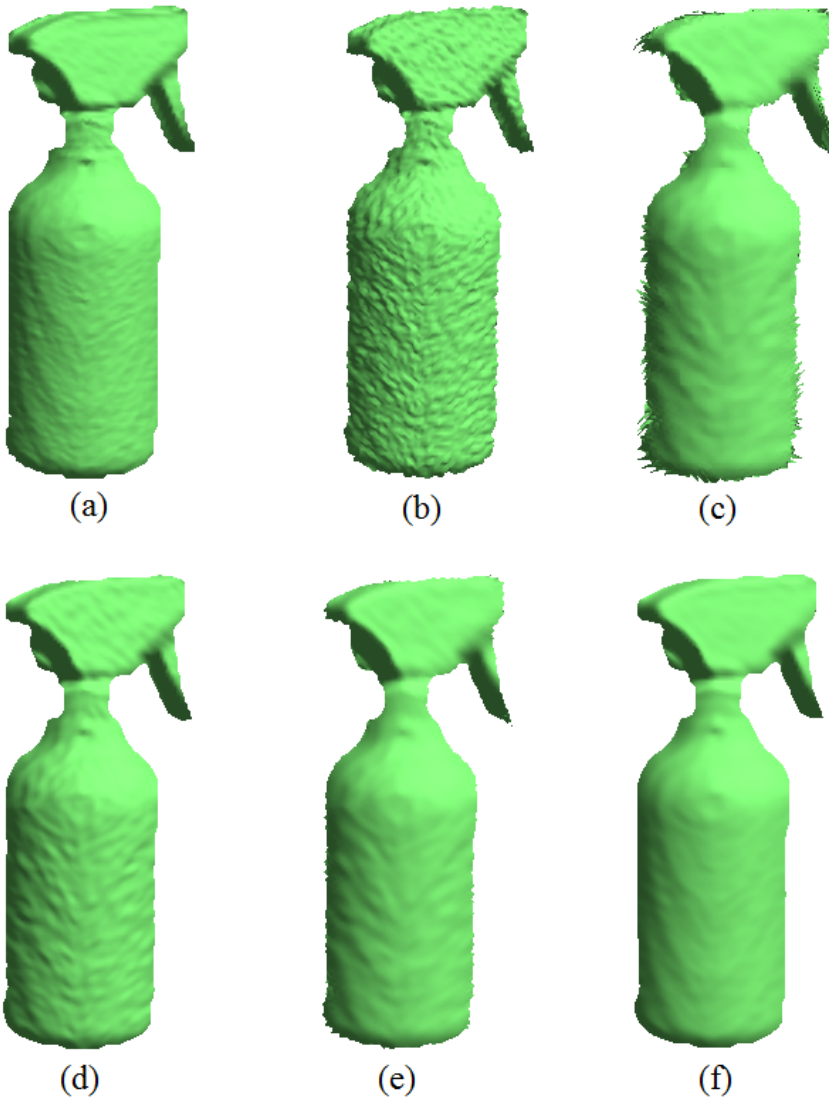


FIGURE 4.8: Smoothing of a noise-added model *bottle*. (a) Original model. (b) Noisy model, with added Gaussian noise, standard deviation=0.2. (c) Bilateral filter results (Iteration times=5). (d) Median filtering results (Iteration times=20). (e) Fuzzy vector median filtering results (First iteration times=20, second iteration times=20, $\sigma = 0.3$). (f) Our result.

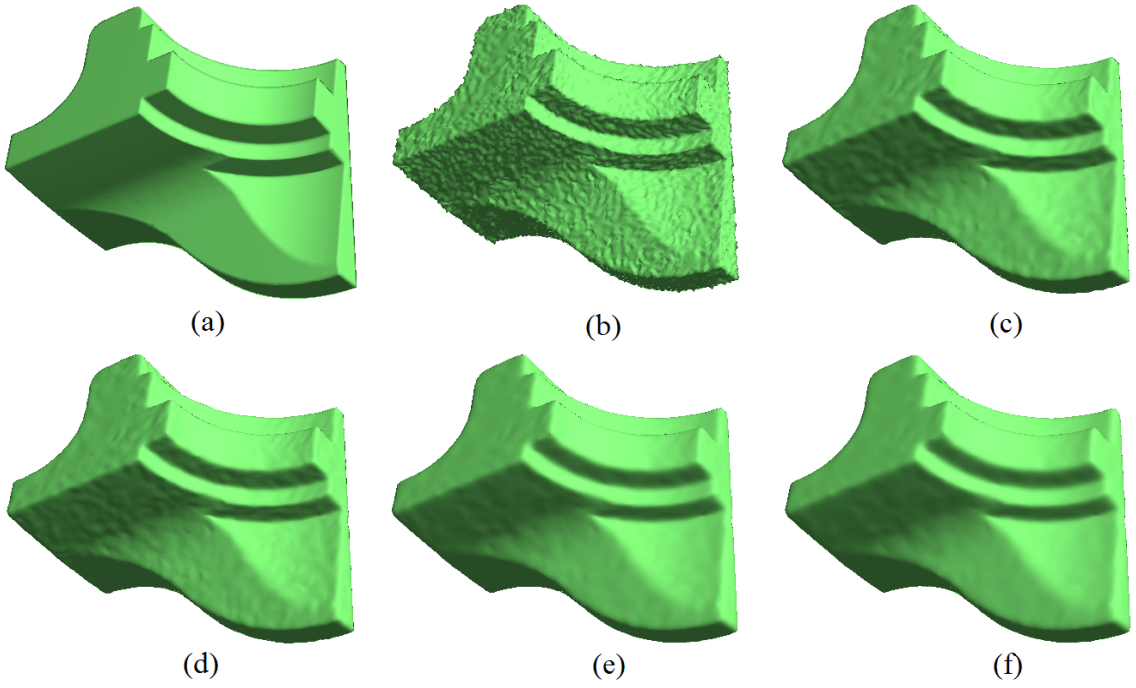


FIGURE 4.9: Smoothing of a noise-added model $fandisk$. (a) Original model. (b) Noisy model, with added Gaussian noise, standard deviation=0.2. (c) Bilateral filter results (Iteration times=5). (d) Median filtering results (Iteration times=20). (e) Fuzzy vector median filtering results (First iteration times=20, second iteration times=20, $\sigma = 0.3$). (f) Our result.

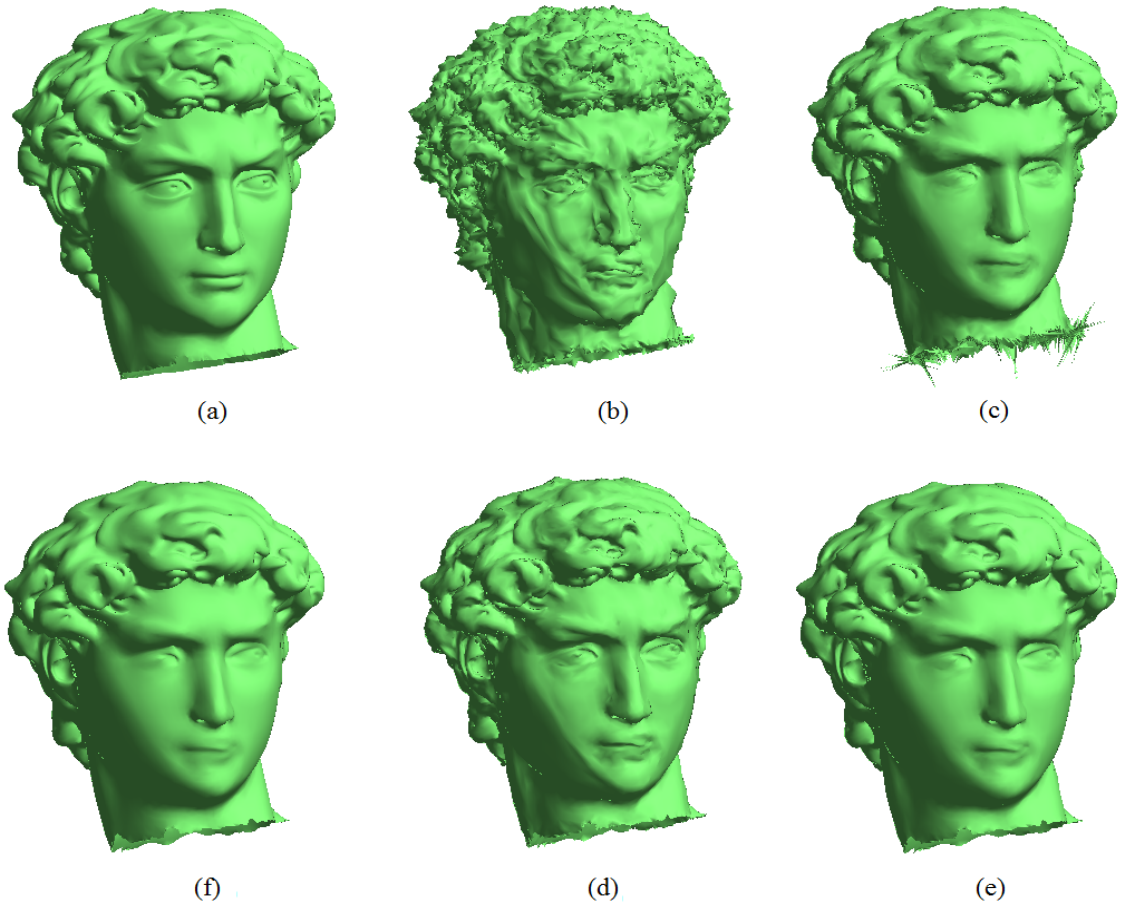


FIGURE 4.10: Smoothing of a noise-added model *david*. (a) Original model. (b) Noisy model, with added Gaussian noise, standard deviation=0.2. (c) Bilateral filter results (Iteration times=5). (d) Median filtering results (Iteration times=20). (e) Fuzzy vector median filtering results (First iteration times=20, second iteration times=20, $\sigma = 0.3$). (f) Our result.

TABLE 4.1: L^2 Error Comparison ($*10^{-3}$):

Method	Fandisk	Buddha	Bottle	David
Bilateral	9.8	9.2	10.5	11
Median	7.4	6.9	8	8.3
Fuzzy	3.5	3.3	4	3.9
Our Method	3.2	2.9	3.6	3.7

4.4.2 Mesh Errors

Many metrics have been proposed in the literature to compare the similarity/difference between two mesh surfaces. These include root mean square error (RMSE), Mean Square Angular Error (MSAE) [248], 3D distance metrics [249], L^2 vertex-based mesh-to-mesh error metric [250], curvature error metrics [251], MESH [252], and Metro [253]. Any one of these may be used as the mesh error metrics. In this section, we have chosen the L^2 vertex-based mesh-to-mesh error metric, as it is frequently cited. In addition, the well-known MSAE error metric is also estimated, as it measures the error from different point view: angle, rather than the distance.

The L^2 vertex-based mesh to mesh error metric is defined as

$$E = \sqrt{\frac{1}{3A(M')} \sum_{p' \in M')} A(p') dist(p', M)^2} \tag{4.15}$$

M is the original mesh, and M' is the smoothed version. p' denotes a vertex in the smoothed mesh M', and $dist(p', M)$ is the L^2 distance between the new vertex p' and a triangle of the reference mesh M which is closest to p'. A is the sum of the areas of all triangles of a mesh in a smoothed version. $A(p')$ is the sum of the areas of all triangles of M' involved with p'.

Table 4.1, above, shows the L^2 errors for the various algorithms considered. In general, our approach produces results almost as good as, or better than those generated by other approaches, whether compared visually or numerically. For instance, the error on model $fandisk$ by our method is only 32.65% of that given by the bilateral filter, and 43.24% of that of the median filter; in model *david*, our method generates an error rate 64.55% lower than bilateral filter, and generates an error rate 53.01% lower than the median filter.

As only small differences are observed in the results of our approach and the fuzzy vector median filtering respectively under measurement by the L^2 vertex-based mesh-to-mesh error metric, we also provide another error measurement: mean square angular error (MSAE) [248]. In some applications, we may only be interested in the error due to

TABLE 4.2: MSAE Error Comparison:

Method	Fandisk	Buddha	Bottle	David
Bilateral	0.2	0.18	0.24	0.26
Median	0.13	0.11	0.21	0.24
Fuzzy	0.05	0.04	0.07	0.1
Our Method	0.01	0.008	0.03	0.05

incorrect direction of our estimated vector. Alternatively, we may wish to have separate characterizations of the bearing error and the error in estimated length [254]. MSAE meets this requirement.

The definition of MSAE is:

$$MSAE = E[\angle(n_{smoothed}, n_{origin})] \tag{4.16}$$

where E is the expectation operator, and $\angle(n_{smoothed}, n_{origin})$ is the angle between the smoothed normal $n_{smoothed}$ and the original normal n_{origin}.

Table 4.2, above shows the large difference between the results obtained by the bilateral filter and our method, as well as by the median filter and our method, which is in line with the result obtained from the L^2 metric. Furthermore, on detailed scrutiny of the results of the fuzzy vector median filter, the differences between the results of this method and those of our proposed method are highlighted: for example, the fuzzy vector median filter error rate is 80% higher than with our method in the *fandisk* case.

4.5 Summary

Many mesh smoothing/denoising methods have been proposed. However, most smoothing methods degrade or remove fine details. In this chapter, we proposed a novel method for smoothing reconstructed surfaces corrupted by scanning noise - to remove noise while preserving mesh features effectively.

The non-local means filter approach has been extensively applied to 2D image processing, and in this chapter, we extended this approach to 3D mesh denoising, and develop a new mesh smoothing method which has a number of important advantages over the main state-of-the-art mesh denoising techniques.

This method is inspired by recent non-local image denoising schemes, is simple and fast, and removes noise effectively, while preserving mesh features such as sharp edges and corners.

The main procedure of this approach is very simple: to measure the similarity between neighborhoods and irregular meshes, we introduced a local approximation utilizing bicubic B-spline surfaces. The new position of vertex of a noisy mesh is obtained as a weighted mean of mesh vertices. Performance with a variety of input data show that our approach can preserve local details of surface.

Comparison then demonstrated that the proposed algorithm outperformed conventional methods in terms of preserving fine details. Overall, all the demonstrated results show that our method preserves surface details better than the other methods. The mean filter may destroy fine features of the surface, and is not feature-preserving. The median filter proved capable of preserving ridge and corner features with a low level of noise, but yielded poor results with a high level of noise. The fuzzy median filter may obtain lower rates of error, and preserve more local detail, but is still inferior to our method. Moreover, the fuzzy median filter takes two steps to compute normal, and uses Gaussian weights, which increases the computational complexity [254].

Through both visual and numerical comparison, it may be observed that the proposed method achieves effective feature-preserving results. The mean filtering approach computes the updated normal of a face using area-weighted averaging of the normals of its neighbours. The limitation of this approach is that it destroys fine features of the mesh, and is not feature-preserving. When there is a high level of noise, the median filter may yield poor results. The fuzzy vector median filter computes normals using two steps: computer a vector median of the normals, and update the normals. Compared to mean filtering approach, which uses the face areas as weights, the fuzzy vector median filter algorithm can yield normals with lower error, but has the disadvantage of using Gaussian weights which increase the computational costs.

Compared to the comparative filtering algorithms, our algorithm is an improvement since it less time-consuming because we do not need to compute the vector median, and our weight function is simpler than its Gaussian weights.

Future work will focus on developing a human visual system perception based metric to evaluated the smoothing results. Moreover, a surface smoothing tool: 3D bilateral filter, recently has been developed (u`http://www.cs.utexas.edu/~bajaj/cvcwp/?page_id=1673`), a comparison between this method and ours will be reported.

Chapter 5

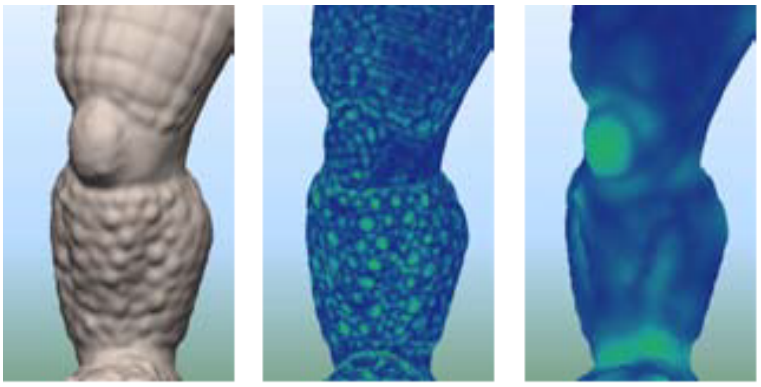
Saliency Detection

With the increasing need for additional detail, the proliferation of digitizing and modelling 3D models has led to a rapid expansion in the number of alternatives. Correspondingly, we are facing a challenge in dealing with massive vertices-contained 3D models - not only for interactive rendering, but also for other surface processing, such as surface simplification, segmentation, shape matching and filtering. To this end, many problems in computer graphics, computer vision, and geometric modelling require reasoning about which regions of a surface are most important. By identifying such important regions, the various subsequent processings of a surface are greatly sped up. For example, in mesh simplification algorithms, the relative importance of vertices may guide the order in which they are decimated; and in an icon generation system, the most important regions of an object should remain visible after processing.

The proposed approach is to compute the surface saliency, in order to analyze the distinctiveness of the region or vertex. This method is based on one or more principles of human visual attention: local low-level considerations, global considerations, visual organization rules and high-level factors [2].

5.1 Introduction

Owing to its efficiency of visual persuasion in traditional art and technical illustrations, visual saliency has now been widely used in computer vision applications, and has been an important pre-processing step for many 3D applications, such as mesh simplification, registration, segmentation and compression. Visual saliency in 2D images is the perceptual quality that makes an object, person, or pixel stand out relative to its neighbours, and that captures our attention [62]. Similarly, the saliency of a 3D surface is the perceptual quality that makes a region, vertex, or component distinctive.



(a) (b) (c)

FIGURE 5.1: Sample to illustrate that the most salient region is not always best assessed from an area of high curvature [3]: (a) *Stanford Armadillo* model. (b) curvature. (c) saliency as detected by [3].

But how can machine vision systems detect the important regions from an unknown input 3D surface? There has been significant progress in the development of algorithms for determining important/ salient regions of 3D surfaces (see the reviews in Chapter 2). Most prior techniques employed purely geometric measures - such as local curvature - or required user input to indicate important areas. For example, Lee et al. [3] defined a measure of mesh saliency using a center-surround operator on Gaussian-weighted mean curvatures. Similar measures of regional saliency have been defined by Shilane et al. [40], Gal et al. [105], Castellani et al. [103], and others. However, in the course of the literature review we found that almost of these methods simply select a region as salient where it presents a large curvature - i.e., where the curvature of a given surface patch is significantly different from that in its immediate neighborhood. These methods mainly employ local analysis and attempt to capture the local variations of the features of the surface without considering the global geometry of the surface.

Intuitively, the salient regions are not those with specific curvature profiles. In other words, strictly geometric measures, such as curvature maxima or minima, do not always correlate with perceptual importance. For example, as seen on the leg of the Stanford Armadillo model in Figure 5.1, the high-curvature spike regions are likely perceived to be important, but are visually monotonous (Figure 5.1(b)). However, the low-curvature knee region will be perceived to be more important as shown in Figure 5.1(c).

A critical question: 'in the case of a 3D surface, which parts are the most salient?'

The salient features of a surface typically characterize the surface well and form a basis for a non-global similarity measure among sub-parts of shapes [105]. In other words, the salient regions are a small number of parts that characterize the object's shape and

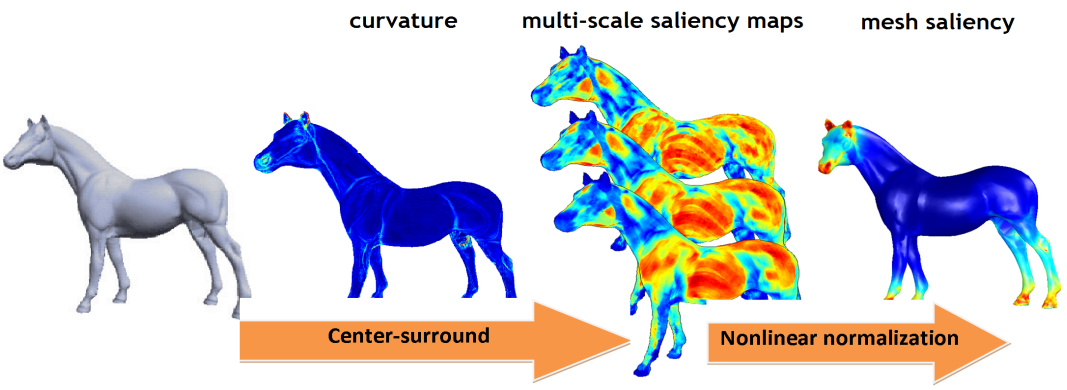


FIGURE 5.2: Outline of the most common saliency computation model, which uses a center-surround mechanism.

details, determined as an index [255]. The detected salient region facilitates the understanding of the structure of a given model, and the identification of regions/components that are particularly important for the specific applications.

Normally, people's attention is attracted by differences: thus, if a region is different from other regions of the surface, we are more likely to say that the region is *interesting* or *salient*. Therefore, we look for vertices or regions that are distinctive in their appearance.

5.1.1 Limitations of The Existing Methods

Figure 5.2 shows the framework of the most common 3D saliency detection methods. Typically, multi-scale saliency maps are obtained by combining curvature information at different scales. While all of the above-mentioned approaches have their advantages, they have the following common disadvantages. Typically, [3, 40, 99] output a single-saliency map by simply computing the sum or the average of all multi-scale saliency maps to simplify the information, and then use thresholding-based methods to determine whether a point is salient or not. While these methods are fast, they do not make good use of the information embedded in the multi-scale saliency maps [111]. On the other hand, a large number of methods, such as [3, 103, 104, 105] rely heavily on the curvature alone. Their performance may degrade catastrophically due to the high sensitivity of curvature to imaging noise. Saliency is a relative concept, judged not just against local neighbours, but also global shape. Essentially, the limitations of existing methods are:

- they mainly consider immediately neighbouring information, overlooking glocal information; and
- they confuse the global overall shape and local details.

5.1.2 Proposed Method

In this chapter, we propose a method for computing saliency that is motivated by surface simplification and interest point detection. The proposed method is a hybrid system, which combines vertex-based and region-based saliency detection, as together they are more likely to characterize both local and global shape properties satisfactorily. Figure 5.3 shows the outline of the proposed saliency detection system. The goal is to calculate salient regions, rather than to calculate saliency only for isolated vertices. This consideration is derived from the observation that the tendency in human perception is to merge close items together.

The proposed vertex-based saliency measure is inspired by Lee's mesh saliency technique. Lee et al. [3] proposed one of the most effective techniques for computing saliency for 3D meshes: *mesh saliency*. By contrast with the case of 2D images, where color is the most important attribute[61], for 3D objects mesh saliency considers the geometry of the meshes as the most important characteristic in generating the saliency. In Lee's work, multi-scale curvature analysis based on a center-surround operator is used: this reliably separates what most observers consider to be interesting regions from the surrounding context, in an entirely automated process. Moreover, *mesh saliency* has been used successfully in graphics applications such as mesh simplification and viewpoint selection. The proposed method differs from *mesh saliency* in making use of two other surface invariants to replace the multi-scale curvature analysis: the Retinex-based importance feature (RIF), and relative distance (RD). As we discussed in Chapter 4, Retinex theory is an image enhancement technique that relies on the model of the human observation of scenes. By adapting this approach, the resulting enhanced shape index map may more faithfully represent both local details and global shape. RD demonstrates that the dissimilarity measure is proportional to the difference in geometrical invariants. Together, these measures are capable of capturing the important aspects of the geometry. Thus, the proposed vertex-based method not only considers the geometry of the surface, but also takes into account the vagaries of human visual perception.

We adopt a strategy that divides the original surface into regions, or patches, to which vertices are attached to aid saliency detection. Most existing of the saliency detection methods normally take into account only local details, whereas our proposed method also incorporates global shape information. This results in more reliable performance. Region-based saliency detection is proposed for stability and continuity in this chapter, as the detection of vertex-based saliency is affected by occlusion and holes. The region-based saliency is a post-processing of a vertex-based saliency map. The methods mentioned above, including the proposed vertex-based saliency method considered in isolation, cannot guarantee completeness of the salient regions: in other words, they

create fragmented regions. In estimating the saliency, they usually only consider local information: as a result, the yielded salient region may lose global information, which sometimes leads to unfavorable results, such as incomplete detection of salient regions. In this work, the regions or patches are determined by surface segmentation. We employed a region-growing segmentation method [146] to segment the surface into patches based on vertex-based saliency values. Finally, the region-based saliency is produced by combining spatial distance and relative importance distance between two selected regions.

Our approach retains the advantages of existing methods [3, 40, 99, 103, 104, 105, 146], such as accuracy. The superior effectiveness of this novel method is finally validated from two independent perspectives. First, we compare our detected saliency with that of other two state-of-the-art detection methods. Secondly, the detected saliency is used to guide two applications: these will be shown in Chapter 6 and Chapter 7, respectively.

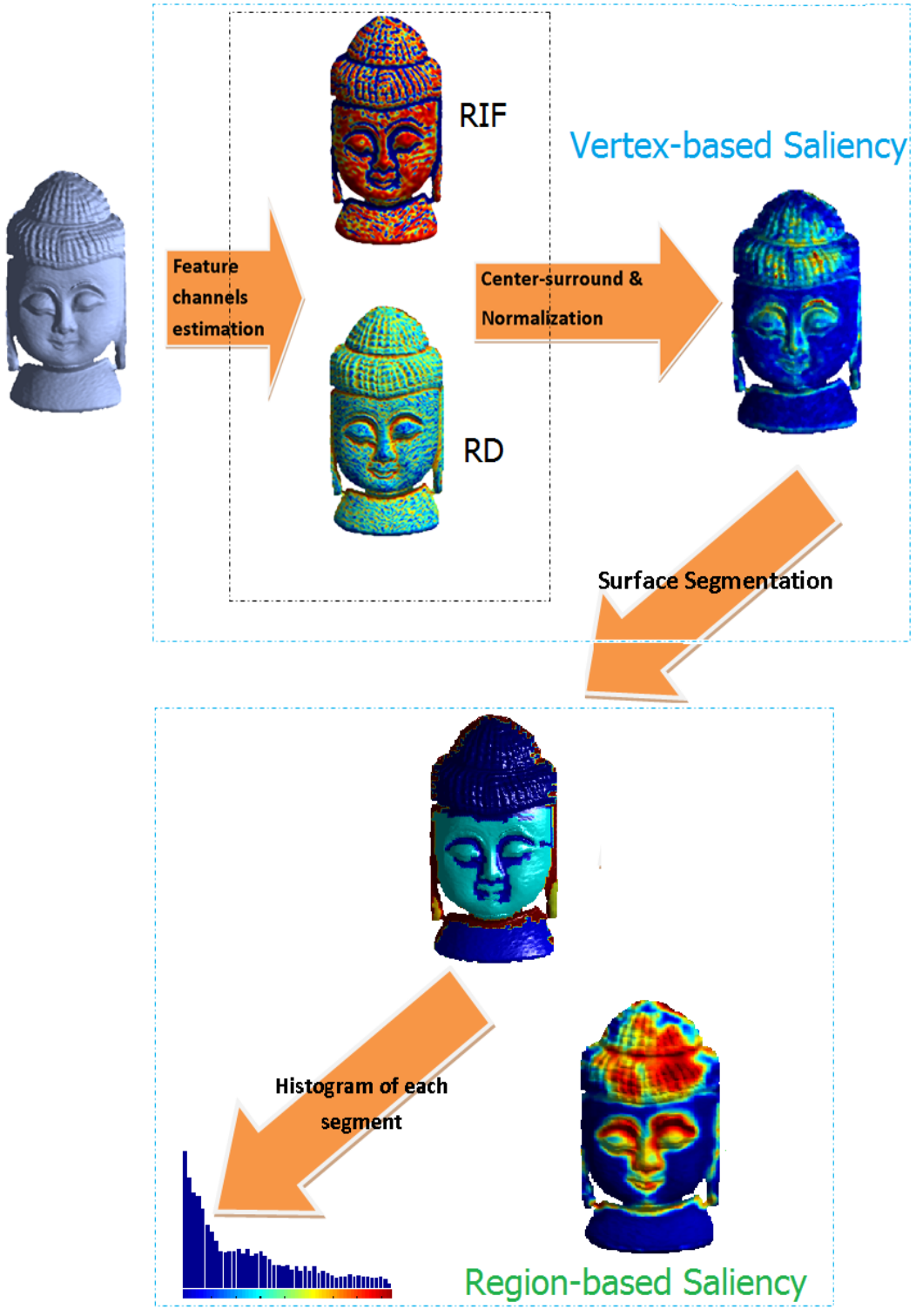


FIGURE 5.3: Outline of the proposed saliency detection system. It consists of two steps: vertex-based saliency detection and region-based saliency detection.

5.2 Vertex-based Saliency

We are seeking to define a saliency map for the surface which is estimated by vertices. A vertex is distinct if its descriptor is dissimilar to all other vertex descriptors of the mesh. In this chapter, we propose a random center-surround saliency map based on the vertices. We improve and extend the concept of *mesh saliency*, integrating into it a better method for neighborhood estimation.

There are several possible characteristics of mesh geometry that could be used for saliency estimation. Lee et al. [3] used the curvature, Shilane et al. [40] employed the discounted cumulative gain and Gal et al. [105] used relative area size and curvature. In this work, the main structure of our saliency system is based on the general layout of psychological attention models. Several surface invariants are investigated in parallel - RIF and RD, and the conspicuities are finally fused into a single saliency map.

Figure 5.4 shows the outline of the proposed vertex-based saliency detection process. As we know, saliency detection requires image/surface invariants in order to guarantee that the detected saliency will not be affected under conditions of image transformation such as changes of scale, rotation, translation and noise corruption. For example, the invariants in the case of 2D images are color, intensity, and relative orientation, whereas for 3D objects the invariants can be curvature, shape index, and some measures derived from the application of Retinex theory.

5.2.1 Relative Distance

In Itti's work, the saliency of a 2D image was estimated by combination of color and orientations, and relative area size was employed to compute saliency in the case of 3D surfaces in [105]. The use of orientation and relative area both reveal the importance of spatial properties for calculating the saliency: the removal of dependence of saliency on the scale and transformation of the object. For this reason, the proposed saliency detection method also employs a surface property from the spatial domain: relative distance. The relative distance is used in case the distribution of the data is not uniform, and the

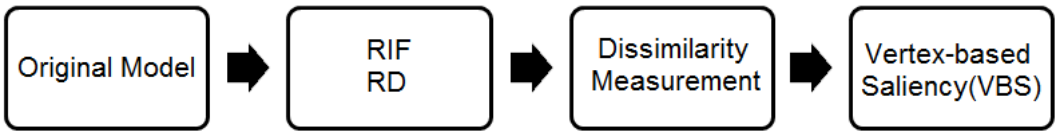


FIGURE 5.4: Outline of the proposed vertex-based saliency detection. The RIF (as outlined in Chapter 3) and the RD are applied to the original surface to estimate the feature maps. The vertex-based saliency is obtained by the dissimilarity measurement.

distance metric mainly focuses on the relationships between neighboring points [194]. For two sets of points with similar neighboring relationships but different densities (i.e., similar relative density), the absolute distances between corresponding points differ dramatically from each other, but the relative distances are in general similar. This is an advantage of the relative distance in reflecting the relative density of points and relative scale of the objects.

In this section, we introduce the definition of the relative distance. Let a point set M contains m points $u_1, u_2, ...u_m$. First, two kinds of relative distance between a pair of points x_i, $x_j \in M$ are defined as follows, according to [194]:

1. **Relative maximum distance:**

$$rd\_max(u_i) = \frac{\|u_i - u_j\|_2}{max_{u_k \subset M}(\|u_i - u_k\|_2)} \tag{5.1}$$

2. **Relative average distance:**

$$rd\_ave(u_i) = \frac{\|u_i - u_j\|_2}{ave_{u_k \subset M}(\|u_i - u_k\|_2)} \tag{5.2}$$

The $max_{u_k \subset M}(\|u_i - u_k\|_2)$ and $ave_{u_k \subset M}(\|u_i - u_k\|_2)$ are the maximum and average Euclidean distance between u_i and other points belonging to M respectively.

Figure 5.5 shows two samples of relative distance values illustrated on the 3D surface. Each of them has the ability to represent the shape. However, on careful observation, the relative average distance produces the more visual pleasing result, and has stronger ability to represent the local geometry information. Therefore, the relative average distance will be used as the relative distance metric for saliency detection in the rest of this chapter.

Note: In our previous work [256], RIF, shape index and curvature were used in combination as the individual feature channels to establish the saliency. However, after careful consideration, such a combination is seen to have a drawback: it may contain a great deal of information redundancy, since each of these channels is based on the principle curvatures. Moreover, the features lack spatial information, and the model may miss detection of salient regions under conditions of spatial transformation, such as rotation. Figure 5.6 presents for comparison the results of two alternative saliency detection methods with the same structure but with different feature channels. Although the previous approach [256] achieved competitive experimental results, we may see that the eyes regions of *buddha* are mis-detected. Consequently , in order to address the main issue of

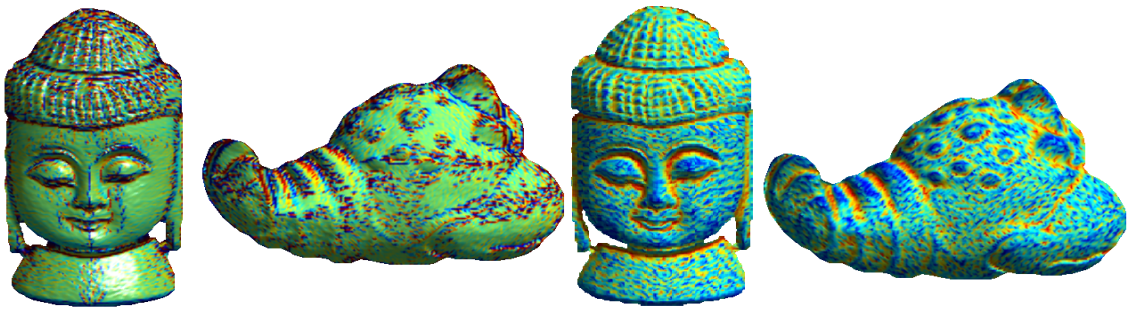
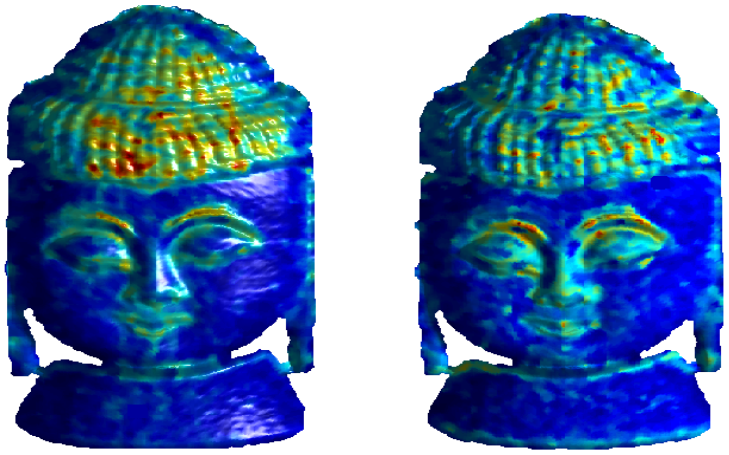


FIGURE 5.5: Examples of relative distance values mapped on two surfaces. Left two images: relative maximum distance. Right two images: relative average distance. The green color indicates the smaller relative distance region, and non-green color illustrates the larger relative distance region.



(a) (b)

FIGURE 5.6: Comparison of saliency detection results with different feature channels. (a) Saliency obtained by the fusion of three feature channel: RIF, shape index and mean curvature. (b) Saliency obtained by the fusion of two feature channel: RIF and RD.

our previous approach, in this dissertation, a new feature, relative distance, is proposed in order better to estimate the saliency, as outlined above.

5.2.2 Dissimilarity Measure

We seek a dissimilarity measure that is robust to small changes on the surface, such as those caused by noise corruption or rotation. As we mentioned above, positional difference is also an important factor: in consequence, the positional difference between two vertices u and v is also calculated, in order to estimate the dissimilarity. This dissimilarity measure is proportional to the difference in geometrical invariants, and inversely proportional to the positional distance. A vertex is less distinct when the vertices similar

to it are nearby, and distinct when similar vertices are far away. Therefore, the dissimilarity measure should be proportional to the difference in appearance, and inversely proportional to the positional distance.

Denotes u, v are two vertices of a given mesh, where the coordinates of the vertices are $u = (x_u, y_u, z_u)$ and $v = (x_v, y_v, z_v)$. The attention values of u due to v in the RIF and RD channels are given by the function $A(RIF, u, v)$ and $A(RD, u, v)$ respectively, where,

$$A(\mathscr{F}, u, v) = \frac{\|\mathscr{F}_u - \mathscr{F}_v\|}{\sqrt{(x_u - x_v)^2 + (y_u - y_v)^2 + (z_u - z_v)^2}} \tag{5.3}$$

where $\mathscr{F}$ is either of the geometric measures of a point, RIF and RD, and $\|\cdot\|$ calculates the Euclidean distance of u and v in the mesh.

After all the attention values in terms of these two kinds of feature are determined, they are fused as attention values $\mathscr{W}$:

$$\mathscr{W} = \sqrt{{A_{RIF}}^2 + {A_{RD}}^2} \tag{5.4}$$

In order to obtain high quality dissimilarity maps for saliency detection, the attention values are further processed with an approach proposed by Achanta [62]:

$$\mathscr{W}' = \|\mathscr{W} - NL\| \tag{5.5}$$

where NL is the non-local means filter-smoothed version of the given mesh that was presented in Chapter 4. Originally, Achanta [62] proposed a frequency-tuned algorithm for computing attention maps that exploits almost all of the low frequency content and most of the high frequency content to obtain high quality saliency maps using color and intensity features. This method treats the entire 2D image as the common surround for any given pixel, and then the attention map is obtained by computing the Euclidean distance of the average CIELAB vector of all pixels of an input image to each pixel of a Gaussian blurred version of the same image:

$$S(x, y) = \|I_\mu - I_g(x, y)\| \tag{5.6}$$

where $S(x, y)$ is the pixel attention value at position (x, y), I_μ is the average of all CIELAB pixel vectors of the image and I_g is the corresponding CIELAB image pixel vector in the Gaussian filtered version (using a 5×5 separable binomial kernel) of the original image. The norm L_2 of the difference is used, since only the magnitude of the differences is of interest, each pixel location is an $[L, a, b]^T$ vector, and L_2 norm is the Euclidean distance.

In our work, we extend this scheme to calculate the attention values of a 3D surface. $\mathscr{W}$ combines the RIF and RD, which can be used to replace the I_μ in Equationeq 5.6. *Lab* color space is designed to approximate human vision: the L component closely matches human perception of lightness. It can thus be used to make accurate color balance corrections by modifying the a and b components, where a and b are the color-opponent dimensions, or to adjust the lightness contrast using the L component. In our method, RIF and RD perform the same functions. The RIF also relies on imitating the properties of human vision.

While a and b are the color-opponent dimensions, they also illustrate the most relevant invariant factor of the image: intensity. The RD metric mainly focuses on the representation of the neighboring relationship between vertices. The RD usually illustrates both local and global geometric information. The RD represents the most important information describing a surface: geometry. Therefore, in a manner of speaking, *ab* and RD have equivalent capability to offer an image or surface.

It follows that the combination of RIF and RD may serve as the most adequate substitute for *Lab* in the 3D domain(the experimental results in the following section will proof this assumption), as it combines both the simulation of human perception and the geometric information.

The dissimilarity may be produced, based on the attention analysis above. Let $d_{position}(u, v)$ be the Euclidean distance between the position of vertices u and v. A dissimilarity measure between a pair of vertices may be given as:

$$diss(u, v) = \frac{\|\mathscr{W}'(u) - \mathscr{W}'(v)\|}{1 + c \cdot d_{position}(u, v)} \tag{5.7}$$

where c is the control parameter, and $c = 3$ in our implementation. The scale of the model may affect this formula, but it will not affect ordering, in other words, the similarity ranking of the vertices stay the same if the scale of the model is changed.

5.2.3 Vertex-based Saliency Computation

We need to compute a distinctness value for each vertex, given the dissimilarity values calculated above. Vertex u is considered salient when it is highly dissimilar to other vertices, i.e., when $diss(u, v)$ is high: $\forall v$. The saliency value of vertex u is defined as

$$Saliency_{vertex}(u) = 1 - exp\{-\sum_{v=1}^{U} diss(u, v)\} \tag{5.8}$$

where U is the total number of the vertices in the given mesh. However, in practice, to evaluate a vertex's uniqueness, there is no need to incorporate its dissimilarity to all other vertices. If the most similar vertices (low dissimilarity vertices) are significantly different from vertex u, then clearly all vertices are also highly different from vertex u. Therefore, for vertex u, we search for the K most similar vertices according to the dissimilarity values, and define the vertex set as $\{q_k\}_{k=1}^{K}$.

Hence, the saliency value of vertex u can be rewritten as

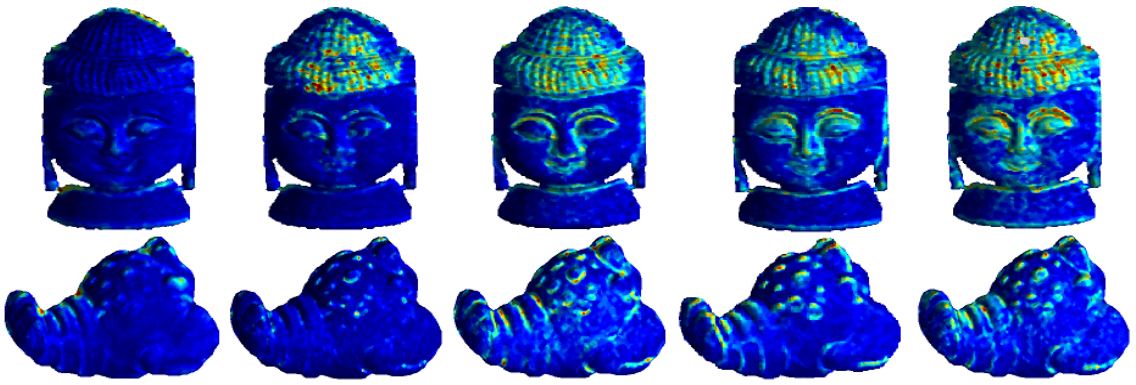
$$Saliency\_vertex(u) = 1 - exp\{-\frac{1}{K}\sum_{k=1}^{K} diss(u, q_k)\}. \tag{5.9}$$

In practice, K is the number of vertices whose dissimilarity value are higher than the average dissimilarity value.

Speeding up the measurement of dissimilarity. Naively evaluating the saliency value for each image pixel using Equation 5.7 takes $O(Num^2)$ time, which is computationally expensive: most of the 3D models we used in this project contain more than 10,000 vertices. In this section, in order to reduce the processing time, an approach aimed at speeding up dissimilarity measurement is proposed. The algorithm is as follows:

1. Input the 3D model data, which is a point set with dimension $3 * W$, denotes:
   - n_1 is a random number where $1 \leq n_1 \leq W$,
   - n_2 is a random number where $1 \leq n_2 \leq W$.
   - Δ is the total random number selection times, and the G is the random number set: $n_1 \in G$, $n_2 \in G$.
2. Compute attention values for both RIF and RD channels: $A(RIF, n_1, n_2)$ and $A(RD, n_1, n_2)$. See Equation 5.3.
3. Determine the attention values S. See Equation 5.4.
4. Estimate the dissimilarity: $diss(n_1, n_2)$. See Equation 5.7

After the processing above, the computational complexity of the dissimilarity measurement is reduced to $\frac{\Delta}{W}[O(Num^2)]$, where Δ is the total number of the random numbers'. Figure 5.7 shows the saliency detection results under different values of Δ: $\Delta = 10\% \cdot W$, $\Delta = 30\% \cdot W$ $\Delta = 50\% \cdot W$, $\Delta = 70\% \cdot W$ and $\Delta = 100\% \cdot W$, respectively, where W denotes the total number of vertices. It may be seen that the higher the number, the more



(a) (b) (c) (d) (e)

FIGURE 5.7: The vertex-based saliency detection results after dissimilarity measurement is sped up, with values of Δ: (a) 10% of total vertex number; (b) 30% of total vertex number; (c) 50% of total vertex number; (d) 70% of total vertex number; (e) 100% of total vertex number.

accurate the detected saliency. As we have performed a large number of experiments, and may also be seen in Figure 5.7 (d) and (e), the salient regions are almost identical when $\Delta = 70\% \cdot W$ and when $\Delta = 100\% \cdot W$. Therefore, for the vertex-based saliency detection, we choose to select 70% of the total number of vertices for the saliency detection.

5.3 Region-based Saliency

In the last section, a vertex-based surface saliency detection was developed, in which the salient regions are detected by a random center-surround saliency model. Although this method is able to capture the regions to which human vision is most sensitive, the experiments still revealed that the method has a drawback: the located salient regions in some cases are not continuous - in other words, it is hard to identify the meaningful regions completely.

It is desirable for the detected salient regions to cover the meaningful regions more completely and faithfully. However, most of the existing methods, including the method we proposed in the last section, detect the salient regions based on the vertices of the given mesh. Because they only consider the local details of the mesh, the yielded salient regions may lack global information. Accordingly, in this chapter we adopt a strategy that divides the original surface into regions, or patches, to which vertices are attached to improve saliency detection. This region-based method considers both local details, and also the global shape information. Thus, it provides a more reliable performance results.

5.3.1 Previous Work Employing A Region-based Approach

First, we will offer a brief overview of prior approaches employing the region/patch technique. Wolf et al. [257] used patches in order to compute texture edges; here the edge representation is powerful in that it can be readily combined with global optimization based-segmentation. The region-based approach has been popular in the field of texture synthesis [258, 259, 260, 261]. The basic use of the patch for texture synthesis consists of stitching together small overlapping patches of the input texture.

The success of region-based methods has been extended to image completion by Drori et al. [262]. This method iteratively approximates the unknown region and fills in the image by means of adaptive fragments.

The region-based approach has also been applied to image denoising. Awate et al. [263] presented several important practical considerations in estimating image-region statistics and in approaching the problem of image denoising. Patch-based methods were also shown to be extremely successful in object recognition [264]. Freeman et al. [265] suggested that region-based representations of 3D objects offer resolution independence over a wide range of scales.

More recently, the notion of a region-based approach has been exploited in the 3D domain. Voshell et al. [266] proposed a patch-based mesh optimization to solve the mesh quality improvement problem by iteratively optimising vertices in each patch of the mesh. Cagniart [4] used the patch-based approach on mesh tracking. A patch-based deformation framework is proposed in this method. Lin et al. [267] gave an adaptive patch-based mesh fitting for reverse engineering. This method segments the input mesh into a set of quadrilateral patches. For each boundary of each patch, it constructs a normal curve and a boundary-fitting curve, which fit the normal and position of its boundary vertices, respectively. Going further, Yamasaki et al. [268] introduced a patch-based compression for time-varying meshes. The patch-based processing proved to be very efficient for eliminating both spatial and temporal redundancy.

In most of the above-mentioned methods, the strategy adopted for the generation of patches/regions is to randomly choose a vertex to be the centre of the first patch, and then to grow this patch until a maximum radius is reached. The subsequent patch centres are chosen among the unassigned vertices that lie on the largest number of patch boundaries. The front of a new patch is propagated from the centre until a maximum radius is reached, or until the processed vertex is closer to the centre of another patch. Figure 5.8 illustrates the patches on the *Stanford armadillo* model generated by using [266].

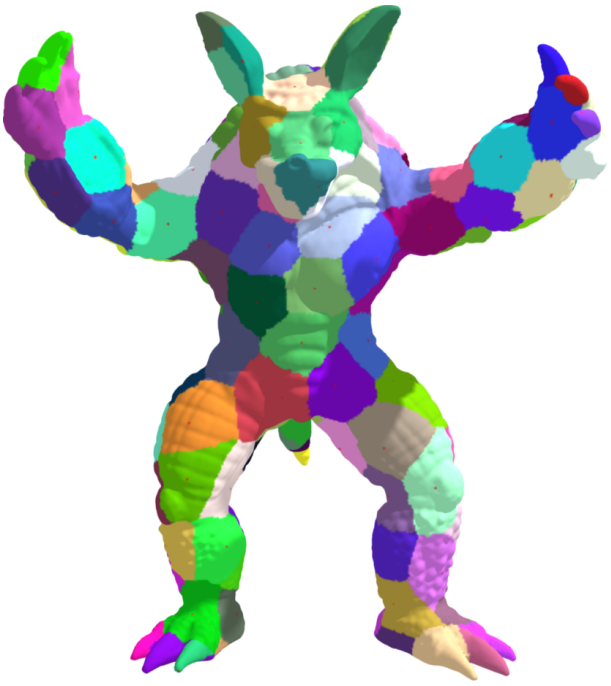


FIGURE 5.8: Patches generated on the Stanford armadillo (173k vertices) with a maximum patch radius of 30 by using [4].

FIGURE 5.9: Outline of the proposed region-based saliency detection. Input with the vertex-based saliency value mapped model. The region is obtained by surface segmentation, in which region growing and graph-based segmentation approaches are combined.

In this work, the surface segmentation is embedded into a framework of the proposed region-based saliency detection. There is a large quantity of literature about surface segmentation: for a recent review and comparison of the most well known mesh segmentation techniques, we refer the reader to [167] and [173]. In this section, 3D surfaces are segmented [146] into a number of regions based on the detected vertex-based saliency values in order to generate the region/patch of the surface for the usage of the subsequent region-based saliency detection process.

Figure 5.9 illustrates the pipeline of the region-based saliency detection process. It may be seen that the VBS values are used as the input data, and the segmentation procedure is required in this framework.

5.3.2 Surface Segmentation

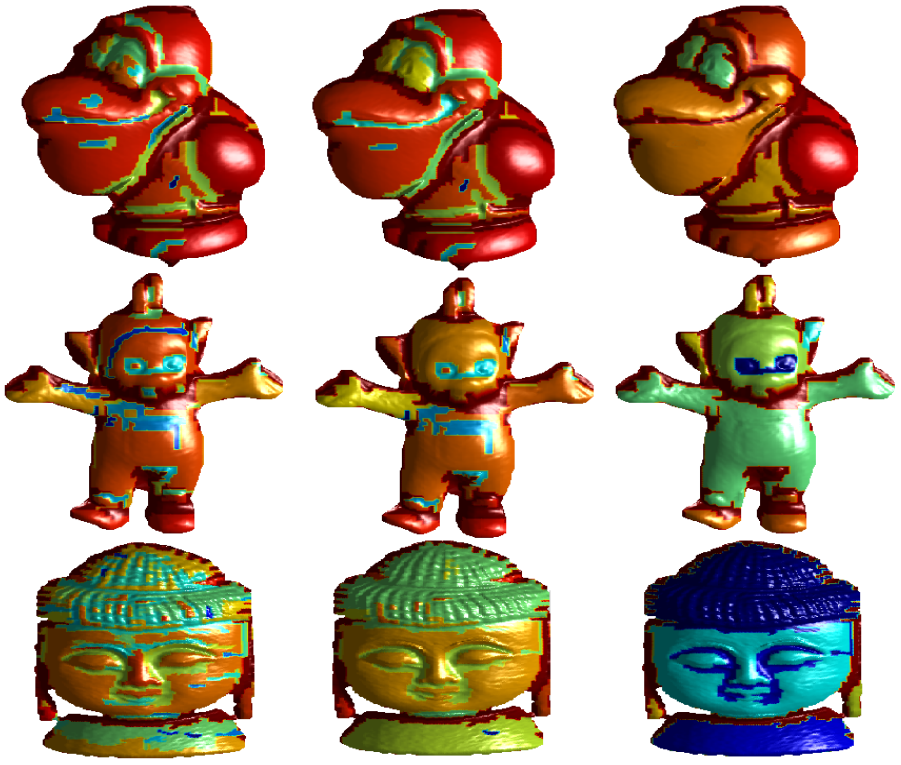
This framework for saliency detection does not need to produce high quality segmentation results: the surface only needs to be segmented approximately. Massive segmentation methods had been proposed, and we have reviewed them in Chapter 2. The region-growing approach [146] is employed in this section for its simplicity and high efficiency. The algorithm for region-growing starts with a seed element and grows a sub-mesh incrementally. Many segmentation techniques are based on region-growing algorithms, and method [146] is employed here because the code is freely available to download online.

More importantly, this algorithm has a region merging process, which is a non-trivial adaptation of an image processing method, taking into account common perimeters of the region. The region-growing scheme is responsible for clustering vertices with similar features. Typically, the vertex with maximal curvature is taken as a start seed. In our case, as the vertex-based saliency has been estimated, the maximal saliency value is used as the start seed. A patch is initialized with the seed and grows from its boundary. If the difference between its saliency value and the average saliency value of the current patch is less than a specified threshold, the growing strategy is greedy in the sense that one adjacent vertex is added only. A new patch is launched by choosing a vertex with maximal saliency from unprocessed vertices until no vertex remains, if no new vertex satisfies the condition. The main difference between various algorithms that use region-growing is the criterion which determines whether an element can be added to an existing cluster. For more details, please see [146].

After vertex region-growing is completed, faces are classified into several kinds according to the saliency values of their vertices. It may be observed that the model is segmented into several segments, and it has the ability to present the the geometrical information that defines the object. Figure 5.10 show the results of the region-growing segmentation with different number of segments.

5.3.3 Region-based Saliency Computation

Humans pay more attention to those image regions that contrast strongly with their neighbors. Besides contrast, spatial relationships play an important role in human attention. High dissimilarity to its surrounding regions is usually stronger evidence for saliency of a region than high dissimilarity to distant regions. If spatial information is introduced directly while computing the vertex levels, a model will consider only local information and may miss global information. In this section, we integrate spatial relationships into regional-level computation.



(a) (b) (c)

FIGURE 5.10: Segmentation results with different thresholds in the region merging process according to the region merging process in [146] (region colors are randomly chosen). (a) 50 segments; (b) 20 segments; (c) 10 segments;

After the 3D surface has been segmented into several regions, we build the saliency model for each region. For a region r_k, the saliency value may be measured by comparing its average vertex-based saliency values to those of all other regions in the mesh:

$$Saliency\_region(r_k) = \sum_{r_k \neq r_i} w(r_i) D_r(r_k, r_i) \tag{5.10}$$

where $w(r_i)$ is the weight of region r_i. Here we use the mean value of the saliency value in segment r_i as $w(r_i)$. $D_r(,)$ is the Euclidean distance of the vertex-based saliency values between two regions:

$$D_r(r_k, r_i) = \sum_{j=1}^{n_k} \sum_{l=1}^{n_i} f(c_{j,k}) f(c_{l,i}) D(c_{j,k}, c_{l,i}) \tag{5.11}$$

where n_k denotes the number of vertex-based saliency values in region r_k, $c_{j,k}$ indicates the j^{th} importance values among all n_k relative importance values in the r_k, and $f(c_{j,k})$ is the frequency of the importance values. All these parameters (n, f) can be estimated by histogram of the selected segment, as shown in Figure 5.11.

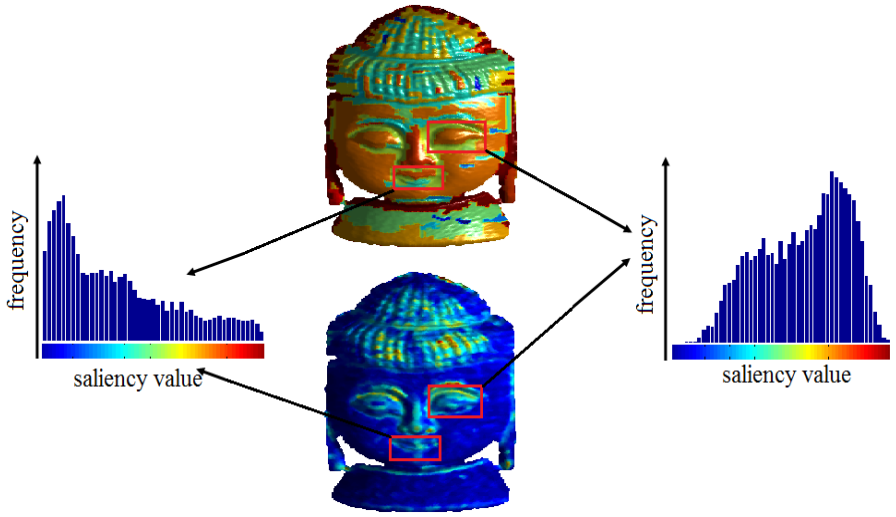


FIGURE 5.11: Illustration of the histograms of the selected segments based on the value of vertex-based saliency.

We further incorporate spatial information by applying a spatial weighting term into the Equation 5.10 to increase the effects of closer regions and decrease the effects of more distant regions. Specifically, for any region r_k, the spatially weighted interim region-based saliency is defined as:

$$Saliency\_interim(r_k) = \sum_{r_k \neq r_i} exp(\frac{-D_s(r_k, r_i)}{\sigma_s^2}) w(r_i) D_r(r_k, r_i) \tag{5.12}$$

where $D_s(r_k, r_i)$ is the spatial distance between segments r_k and r_i. The spatial distance between two segments is defined as the Euclidean distance between the centroids of the respective segments. σ_s controls the strength of spatial weighting. Larger values of σ_s reduce the effect of spatial weighting, so that more distant segments would contribute more to the saliency of the current segment. In our implementation, we use $\sigma_s^2 = 0.4$, and $w(r_i)$ is the weight of segment r_i, as the average saliency value of the vertices in the segment.

However, as we can see that the interim saliency is region-based, which means each segment (region) only has one saliency value, leading to mosaic-like results. Therefore, for more visual pleasing saliency results, a normalization on the interim saliency is provided, which is based on the previously proposed vertex-based saliency. The the final saliency value for the 3D surface can be defined as:

$$Saliency\_final = \frac{Saliency\_interim}{Saliency\_vertex} \tag{5.13}$$

The saliency values are normalized into the range of $[0, 1]$ finally for the convenience of further saliency-guided applications.

5.4 Experimental Results and Discussion

We have developed a model for saliency detection on 3D surfaces, and discussed its computation. In this section, to illustrate which regions are detected as salient by the proposed methods, we show the results of saliency detection by vertex-based saliency (VBS) and region-based saliency (RBS) detection separately. In all figures, regions shown in warm colors (non-blue) are the salient regions.

Figure 5.12 shows the results of the VBS algorithm for a representative model. It may be seen that the VBS algorithm usually detects the 'expected' region of interest. For example, the whole eye region is detected as salient, as the eye is unique to *duck*. For the model *frog*, our algorithm detects the eye, mouth, nostril, and toes as interesting regions: similarly, in the *doughboy* model, the face, cap, and scarf are also detected as salient.

It is worth noticing that the hair of *buddha* is also highlighted as salient. For some methods, such as mesh saliency, repeating patterns are usually ignored for the purposes of saliency detection. In the case of *buddha*, the hair is highly curled, and it is a very significant characteristic of the model. The VBS algorithm has the ability to detect such features as salient. However, on careful observation, some salient regions are discontinuously presented: instances include the eyes of *rick-face* and *pat-face*, where parts of the eye regions are not detected as salient.

In contrast, the results of the RBS in Figure 5.13 are more accurate than those of VBS estimation: RBS detects the salient regions more completely. Region-based saliency estimation has the ability to characterize both local details and global shape accurately, since it uses the patch details rather than single vertices. The region-based method is powerful for the detection of salient regions, successfully combines local and global information, and is robust to imaging noise.

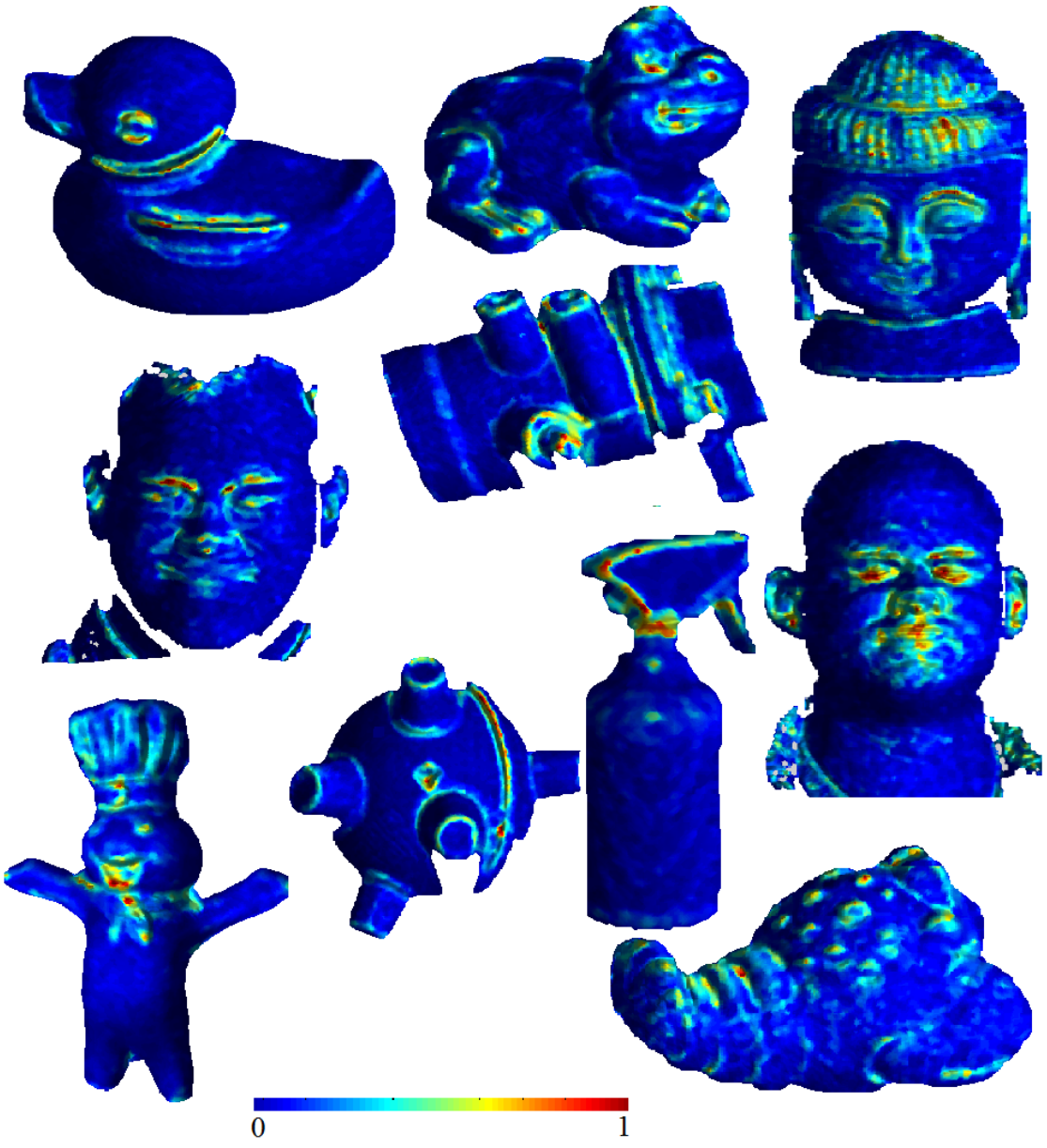


FIGURE 5.12: The results of vertex-based saliency (VBS) detection on different models. The warm colors (non-blue) indicate the salient regions. In all cases, the red regions are the most salient, blue the least salient.

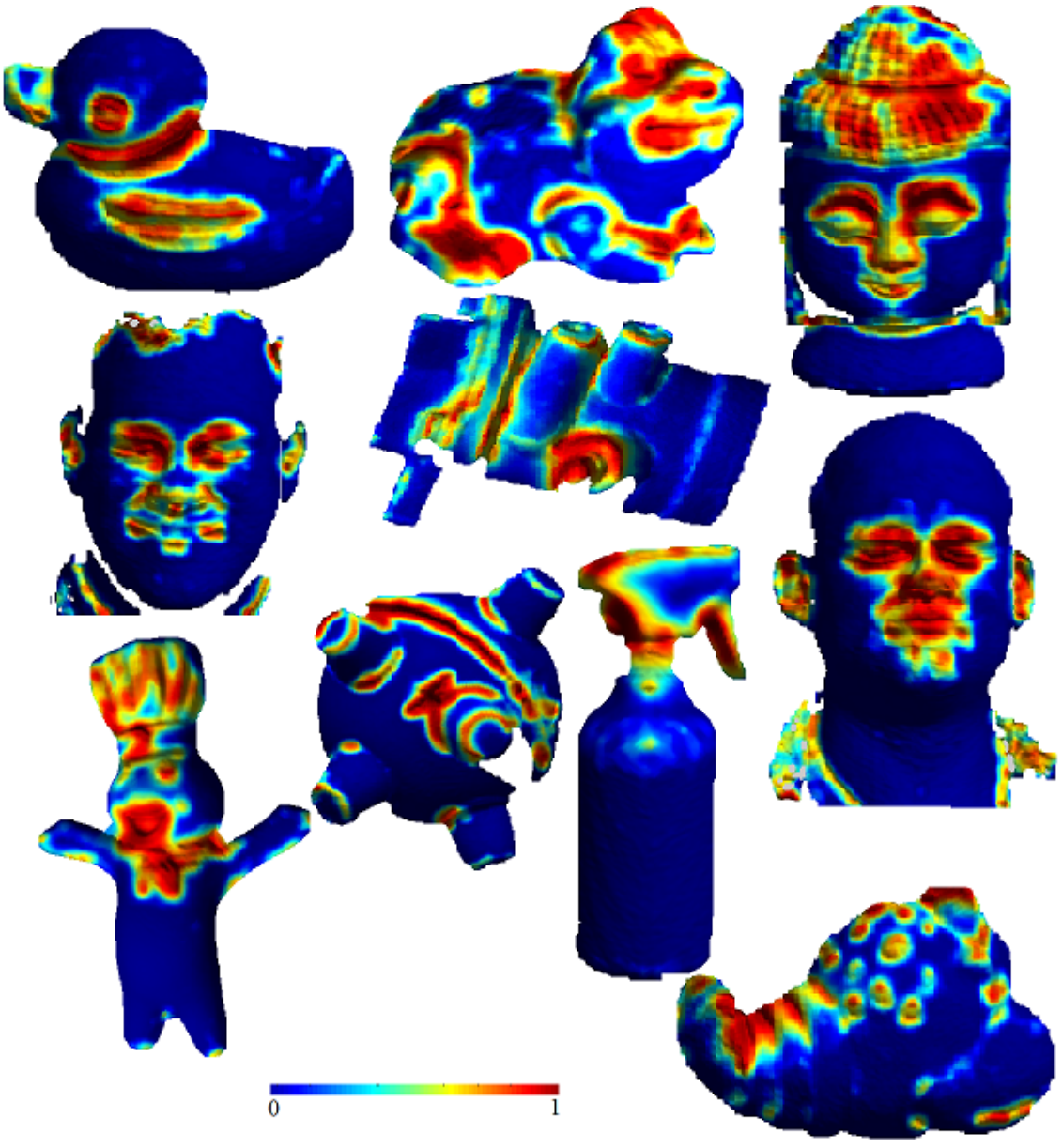


FIGURE 5.13: The results of region-based saliency (RBS) detection on different models. The warm colors (non-blue) indicate the salient regions. In all cases, the red regions are the most salient, blue the least salient.

5.4.1 Effects of Viewpoint Changing

A key feature of the proposed methods for saliency detection is that the results are independent of the viewpoint. In this section, we investigate how the salient regions of a surface are resistant to changes of the viewpoint.

Figure 5.14 demonstrates the robustness of the VBS method under rigid transformations (rotation and translation). The models are rotated 20° and 40° respectively around an unknown rotation axis, followed by a translation. As we can see, the salient regions are retained constantly; in particular, the eyes of *buddha*, the bumpy areas of *lobster*, and the wing of *duck*. Figure 5.15 illustrates the RBS method under rigid transformations. Once more, the models are rotated 20° and 40° respectively around an unknown rotation axis, followed by a translation. The salient regions are retained constantly as well as under VBS. The salient regions such as the eyes of *buddha*, the bumpy areas of *lobster* and the wing of *duck* are completely detected.

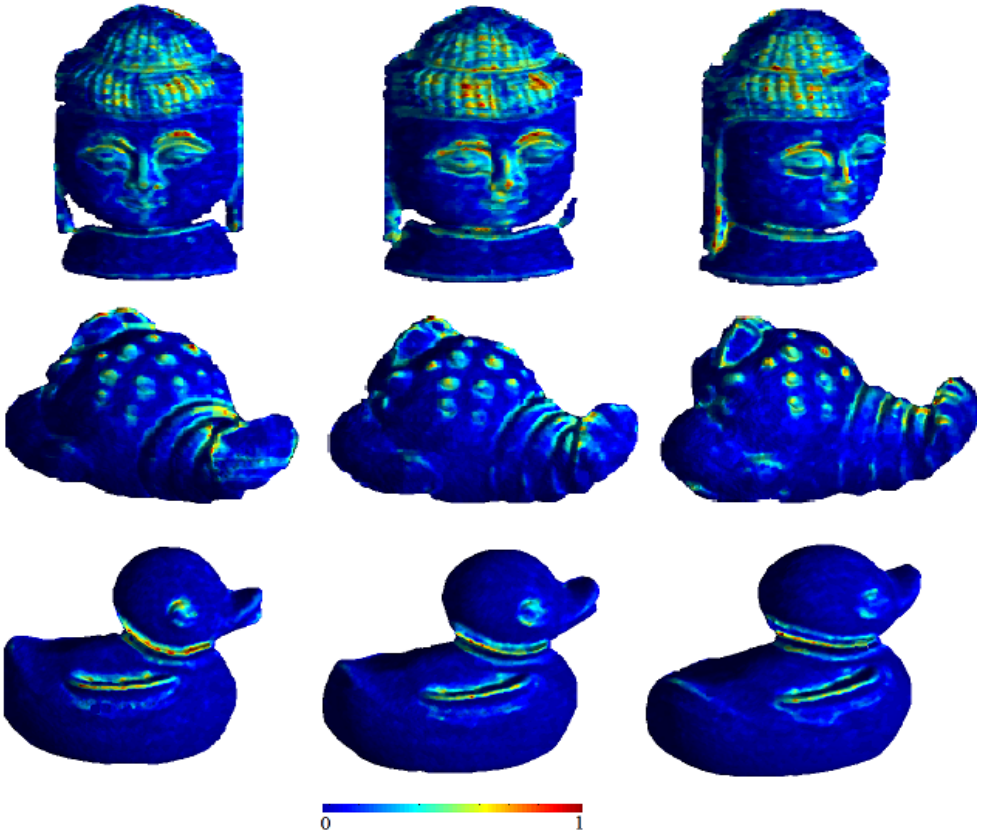


FIGURE 5.14: Visualization of salient regions (warm colors) on different models by the VBS method. The models are rotated 20° and 40° respectively around an unknown rotation axis. From top to bottom: *buddha*, *duck*, and *lobster*: (a) reference viewpoints; (b) 20° rotated; (c) 40° rotated.

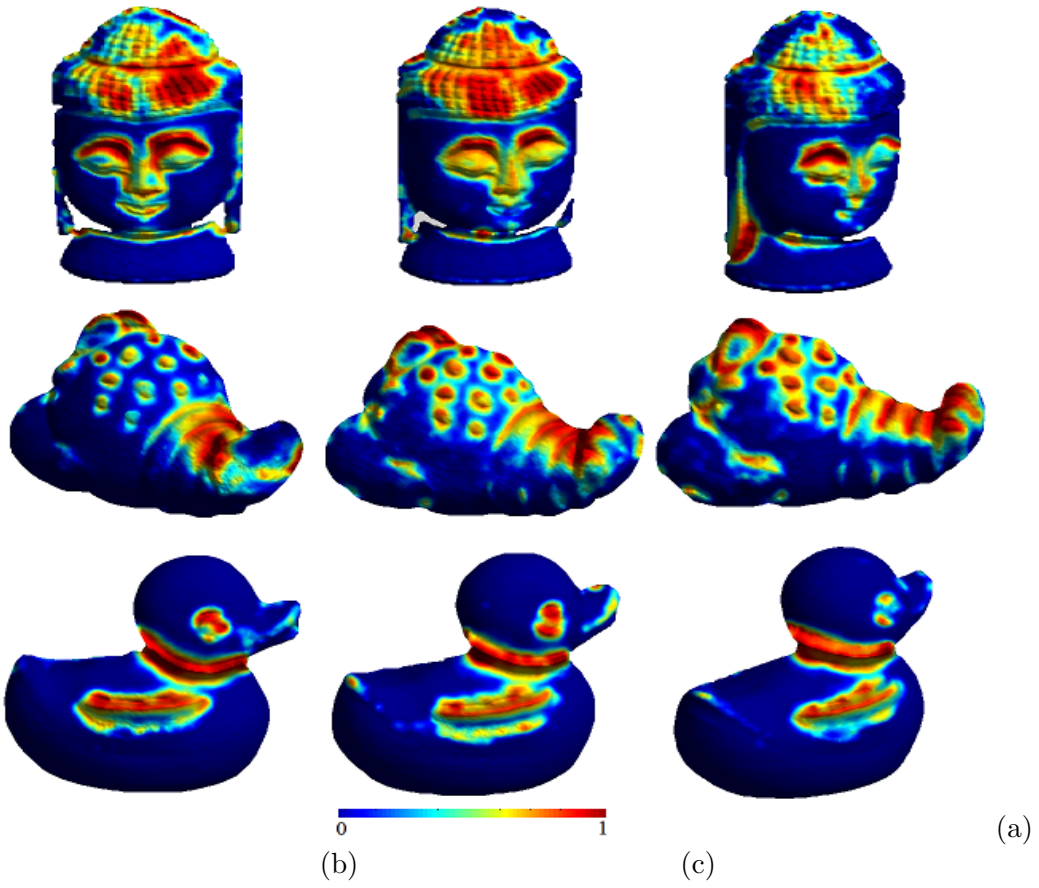


FIGURE 5.15: Visualization of salient regions (warm colors) on different models by the RBS method. The models are rotated 20° and 40° respectively around an unknown rotation axis. From top to bottom: *buddha*, *lobster*, and *duck*. (a) reference viewpoints; (b) 20° rotated; (c) 40° rotated.

5.4.2 Effects of Noise

To demonstrate the robustness of our methods, we also added random Gaussian white noise with standard deviation of σ to the original data, and then ran our method again. The noise made the surfaces more discontinuous, because its usual effect is to introduce roughness. As a result, the detected salient regions of a noisy surface may differ from those of the original.

The results of VBS saliency are shown in Figure 5.16, and RBS saliency are shown in Figure 5.17. It may be seen that both of our methods can still distinguish salient and non-salient regions in the presence of a considerable amount of noise. Broadly, it may be observed that the salient regions estimated from noisy data and those derived from clean data are very similar. In the cases of models *buddha*, *duck*, and *valve*, the detected salient regions remain resistant to noise for the purposes of saliency detection. These results show that either globally random sampling or patch-based method are robust to random noise, and our methods are powerful in detecting salient regions from surfaces with varying complexities of geometry.

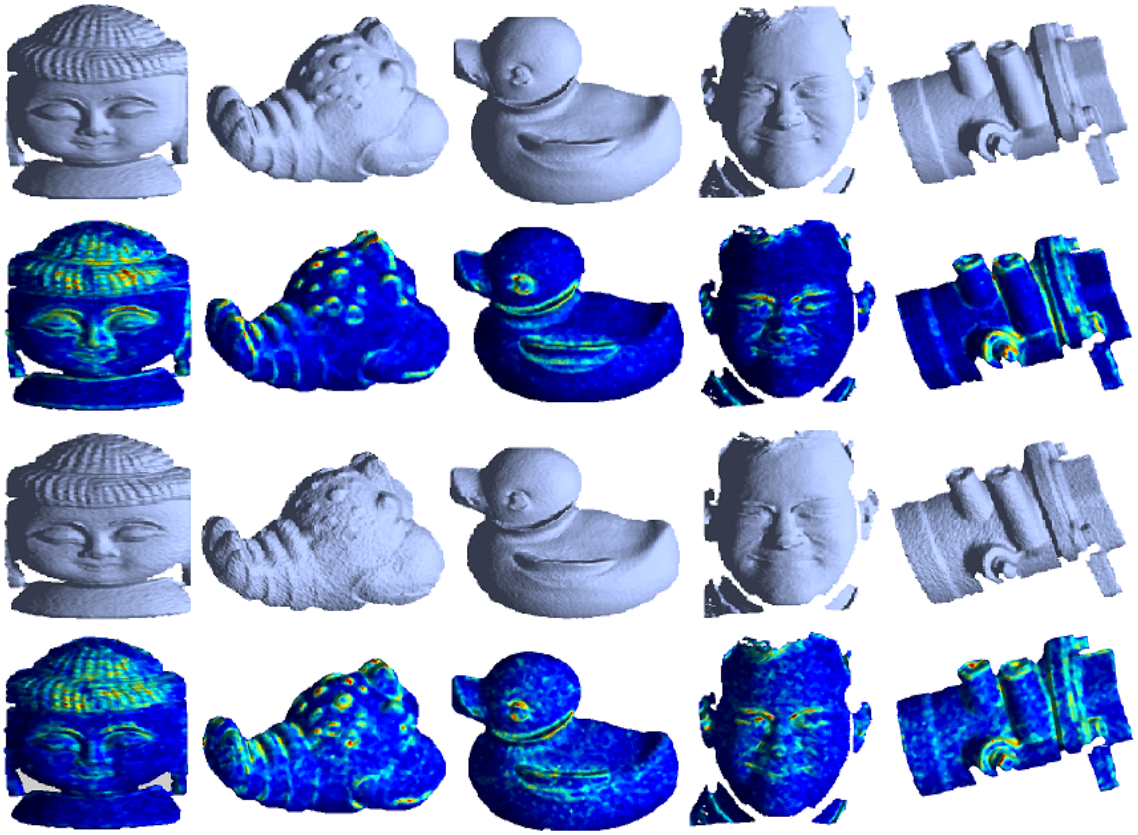


FIGURE 5.16: Illustration of the robustness of the proposed VBS method, tested by adding random Gaussian white noise, $\sigma = 0.3$. From top to bottom: the original surface; VBS-detected salient regions; the noise-added surface; VBS-detected salient regions over the noise-added surface.

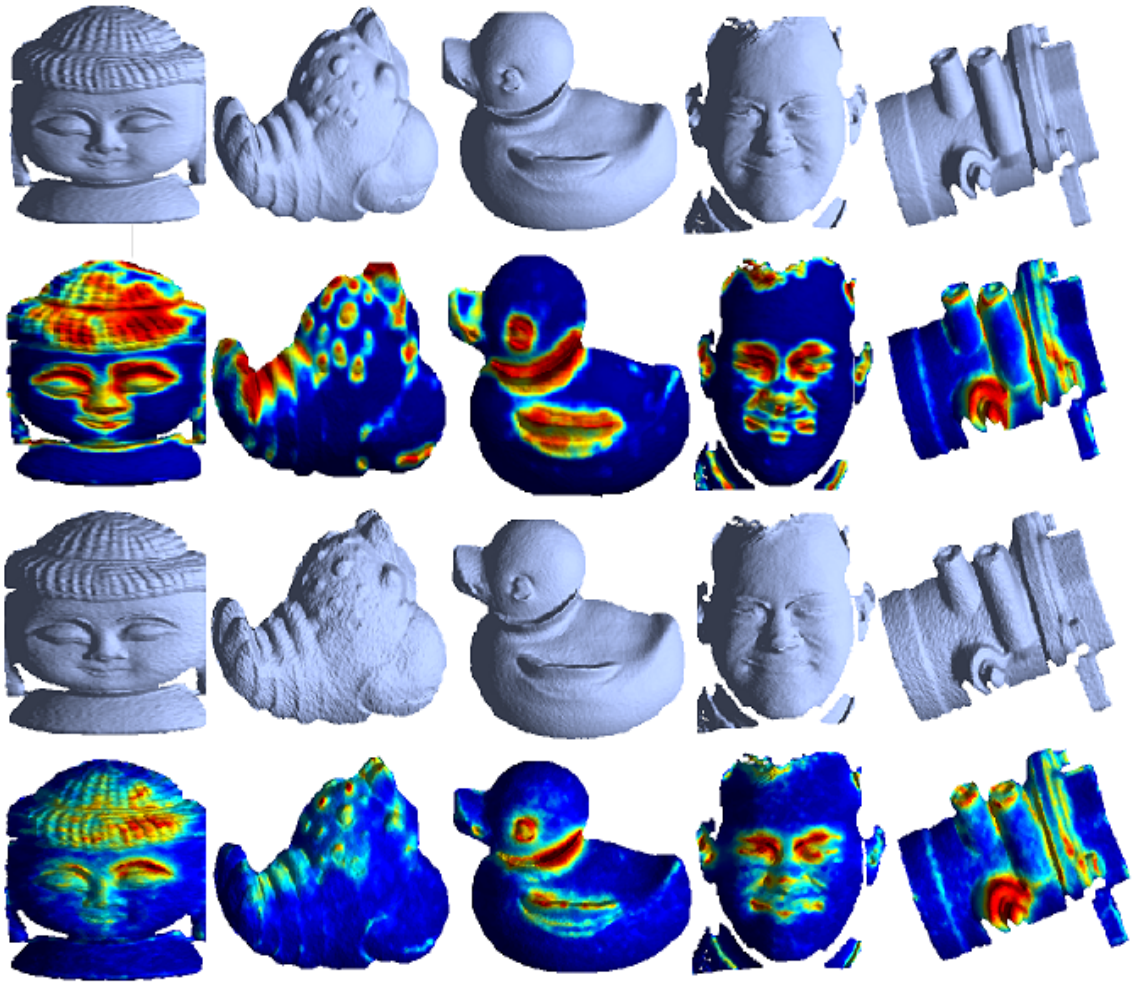


FIGURE 5.17: Illustration of the robustness of the proposed RBS method, tested by adding random Gaussian white noise, $\sigma = 0.3$. From top to bottom: the original surface; RBS-detected salient regions; the noise added surface; RBS-detected salient regions over the noise-added surface.

5.4.3 Effects of Points Sampling

Accuracy and computational cost in our method depends on the number of vertices in the input model. Normally, as long as the number of vertices retained after simplification is sufficiently large, the number has no significant effect on saliency detection. Therefore, in this section, the robustness of simplification will be demonstrated.

We sampled points with different rates - 0%, 50%, 80% and 95% respectively - from models *duck*, *lobster*, *rick* and *buddha*, in order to construct progressively simplified models. If the sampling rate is 95%, this means that only 5% of vertices are retained. The surface sampling was implemented through the Finite Element Sampling Method (FEM) [269]. This method is fast, and sensitive to local features.

Figure 5.18 and Figure 5.19 show the results of the proposed VBS and RBS saliency detection methods when applied after the simplification of surfaces. The distribution of the detected salient regions on each simplified surface does not change extremely, even with a sampling rate of 95%. Such features as the eyes, nose, mouth and hair are fully detected in the simplified version of *buddha* with 50%, 80% and 95% sampling rates. Under the same conditions, the bumpy areas are also located as salient on *lobster*. However, as expected, large distortions are obtained with a 95% sampling rate, since so few points are insufficient to depict the original geometry and details. These results indicate that the proposed saliency detection method is robust on simplified surfaces.

The advantage of such performance is that simplification may be used as the first step in saliency detection for surfaces containing a large number of vertices, without significantly compromising the results of subsequent processing steps.

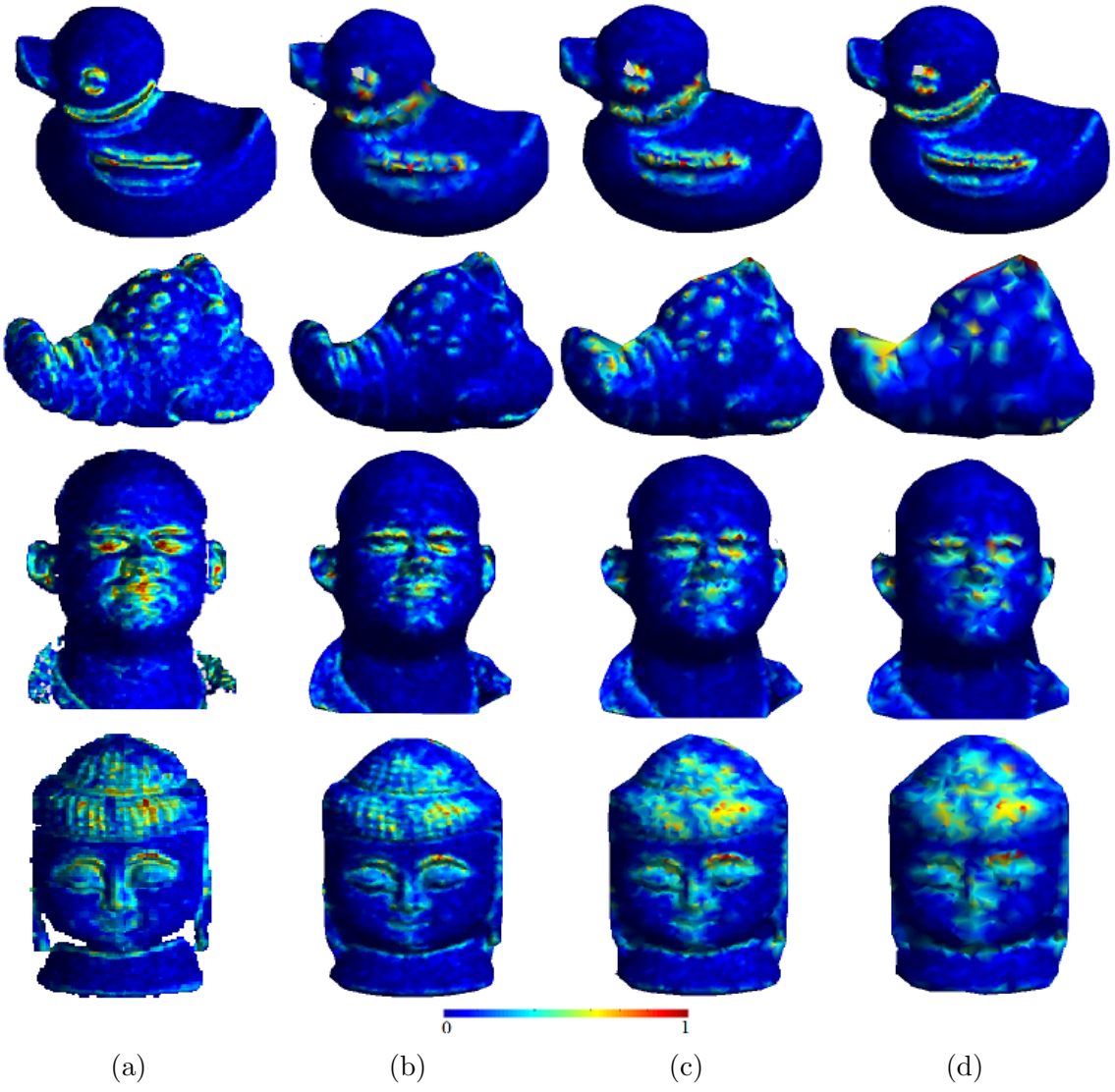
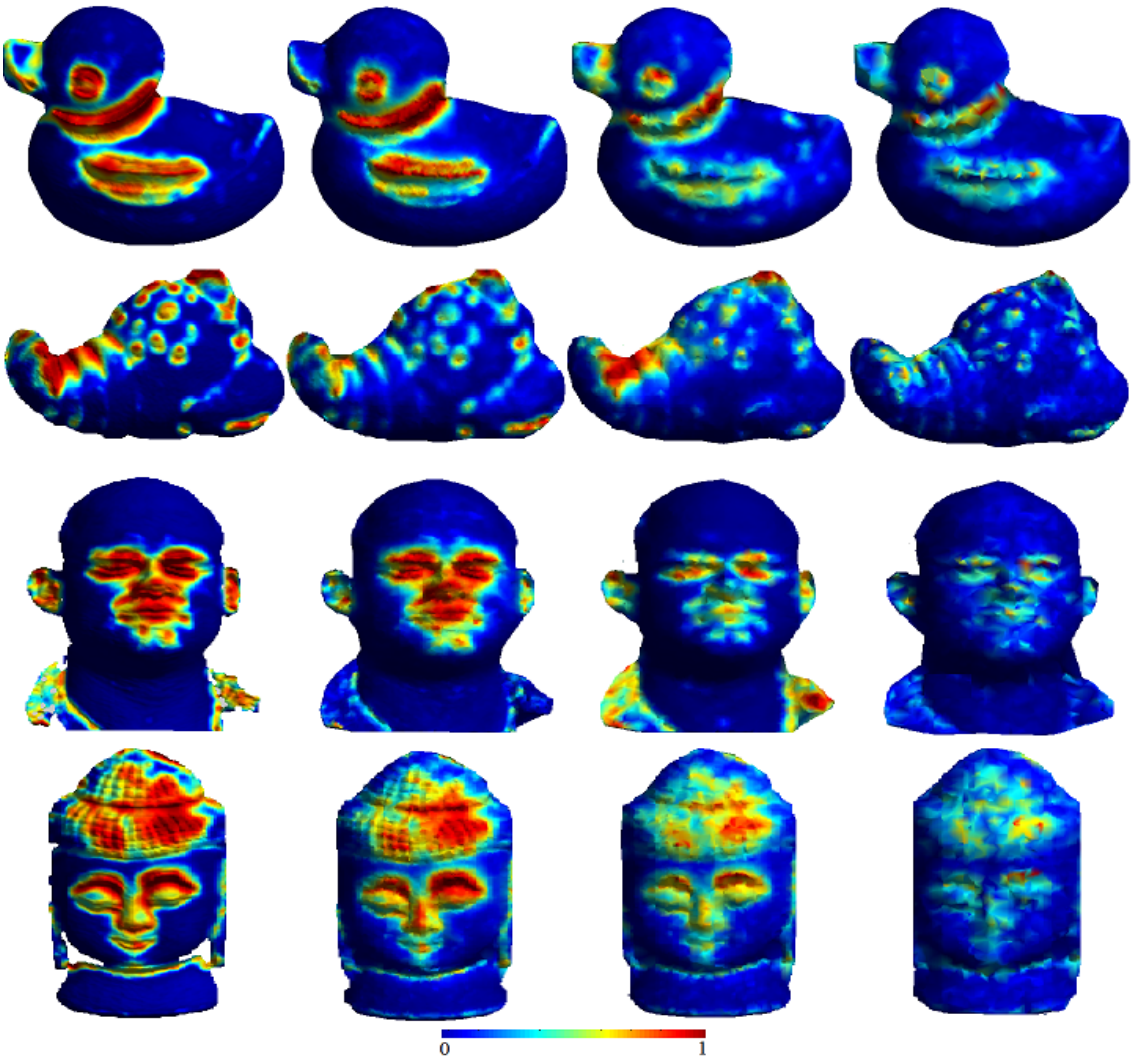


FIGURE 5.18: The proposed VBS method on simplified surfaces with different simplification rates: (a) the original; (b) 50%; (c) 80%; (d) 95%.



(a) (b) (c) (d)

FIGURE 5.19: The proposed RBS method on simplified surface with different simplification rates: (a) the original; (b) 50%; (c) 80%; (d) 95%.

5.4.4 Visual Comparison

We continue by comparing the saliency detection results of four different methods, two of which are the methods proposed in this chapter: vertex-based saliency detection (VBS) and region-based saliency detection (RBS). The other methods include two state-of-the-art methods: Mesh Saliency by Lee et al. [3], and Distinctive Region by Shilane et al. [40]. These methods are referred to as VBS, RBS, MESA, and DIRE, respectively. The warm (non-blue) colors indicate regions of greater saliency.

Figure 5.20 shows a visual comparison of detected salient regions under different saliency detection methods. It may be seen from the results that MESA can capture the visually salient features and locate the large curvature regions of the *lobster*, *buddha*, *duck* and *frog* surfaces, but fails to describe the salient and important regions in the *rick-face*, *valve*, *frame* and *pat-face* models: the original models contained significant noise, and the definition of saliency as a function of the mean curvature is sensitive to such noise.

The DIRE produced relative better results than MESA. For some cases, such as *rick-face*, the salient regions are over-detected: not only has it detected the eyes, nose and mouth, but it has also located the forehead and some parts of cheek as salient regions. The hair region of *buddha* is not completely detected, while only the high curvature areas are defined as salient. The boundaries of the shape are detected as salient regions.

In the case of VBS, the salient regions are distributed in the high geometrical information area, but the located salient regions are not continuous. For instance, on the *buddha*, the nose and eyes are detected incompletely. VBS hardly describes the nose, eyes and mouth area of model *pat-face*: because this method uses a random number of vertices to calculate the saliency, it may have lost some vertices information.

The last column of Figure 5.20 shows the results of the RBS. This method addresses the drawbacks of all of the above methods: salient regions falsely or partially detected, or missed; and over-sensitivity to noise or boundaries. It may be observed from the results of the proposed method that the salient regions are expanded, and correspond to the components of the object of interest. The 'warm color' has covered the most important regions (visually and geometrically) of the surfaces.

In summary, our method achieves more visually pleasing results. The visually important and geometrically distinct regions of the given 3D surface have been detected faithfully by the proposed RBS method, because it estimates saliency based on the average values of patches, rather than on a single vertex.

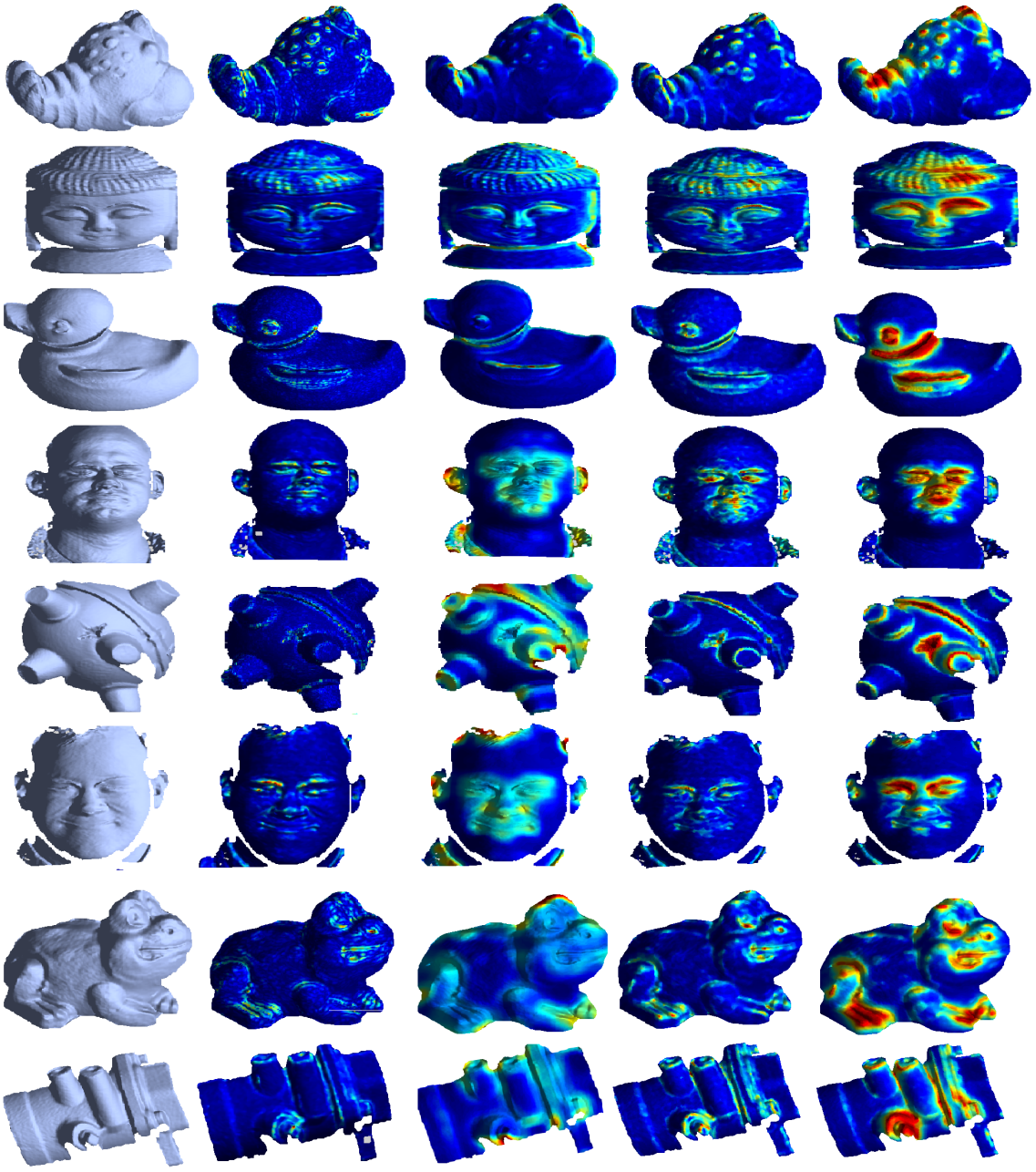


FIGURE 5.20: Visual comparison of detected salient regions under different saliency detection methods. The non-blue areas indicate the more salient regions. From left to right: the original, MESA, DIRE, VBS, and RBS. Top to bottom: models *lobster*, *buddha*, *duck*, *rick-face*, *frame*, *pat-face*, *frog*, and *valve*,.

5.5 Summary

In summary, we have studied the salient regions of a 3D surface, and two considerations have been discussed: vertex-based saliency and region-based saliency. Most of the existing methods use several possible characteristics of mesh geometry to estimate

the saliency, such as curvature. However, such a feature is vulnerable to influence by an external factor, i.e., noise. Therefore, the VBS adopts the RIF and RD as feature channels, and by uses the random center-surround mechanism to estimate the saliency. RIF is the feature based on an image enhancement technique, which relies on the model of human observation of scenes. The shape index of the surface may more faithfully represent both local details and global shape by adopting this approach. As we know, the methods [61, 105] use spatial properties for calculating the saliency of a 2D image and 3D surface, respectively. Thus, we have also used a spatial property: relative distance (RD). RD demonstrates that the dissimilarity measure is proportional to the difference in geometrical invariants.

The experimental results of VBS are acceptable and the method performed rather better than the existing techniques. However, it also revealed a drawback: under VBS, the located salient regions are not continuous, because the dissimilarity calculation used only the local information, and lacked the global details. For this reason, the region-based saliency (RBS) was proposed. First, the surface is segmented into patches by a segmentation approach. Finally, the region-based saliency is produced by combining spatial distance and relative importance distance between two selected regions.

These two saliency definitions produce measures of salient regions under transformations of different types, such as rotation, translation, noise corruption, and degree of points sampled. For a number of examples, we have shown that the most salient parts/regions are consistently detected in a more stable fashion with our proposed methods.

Chapter 6

Saliency-guided Simplification

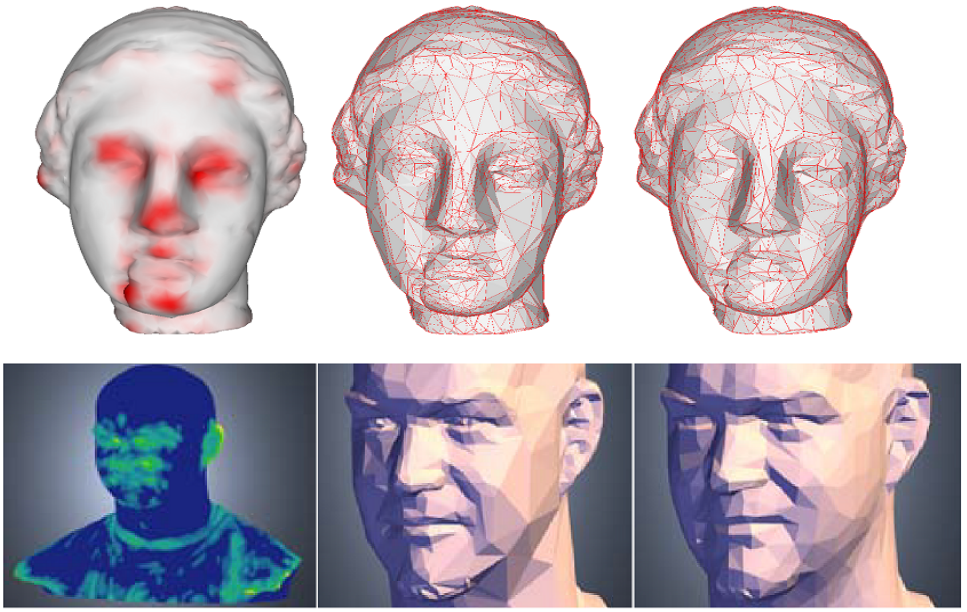
Saliency detection, as a technique in 3D surface modelling, is of broad interest, since it can potentially improve the perceived quality of the results of many 3D surface processing applications. The applications of saliency detection to 3D surfaces are usually fairly straightforward. In many cases, the applications can be performed by use of a saliency map as a weight map. Typical saliency-guided applications are: simplification; levels of detail; viewpoint detection; interest point detection; shape matching; mesh resolution enhancement; and point cloud registration.

In this chapter, we present quantitative comparisons through the root mean square error (RMSE) and the Metro error to demonstrate that simplification guided by our saliency performs better than its competitors. Results show that our methods improve both qualitatively and quantitatively on earlier methods.

6.1 Introduction

The first application in which usage of saliency is illustrated is simplification of 3D surfaces. This application is based on the idea that a higher saliency value for a given vertex indicates that it resides on a computationally more interesting region of the mesh. Thus, the goal of saliency-guided applications is to delay the simplification of the salient parts of the surface, while strengthening simplification of the non-salient regions.

As more computational power is not always available, highly detailed models must be simplified in order to be displayed interactively: the major challenge is to maintain the visual fidelity of the models under simplification. However, geometrically-based simplification risks making further processing pointless, if the distinguishing characteristics of



(a) (b) (c)

FIGURE 6.1: Examples of saliency-guided simplification results generated by two different methods: (a) saliency maps; (b) saliency-guided simplification results; (c) Garland's method (Qslim) [1]. Top row: *Schelling points* method [77]. Bottom row: *Mesh saliency* method [3]. It is clearly shown that the important regions have been well-preserved: the eyes and noses from two different models are better preserved by using saliency-guided simplification methods than traditional geometric-based simplification method.

the object are rapidly lost. One promising solution would be to use perceptually adaptive graphics, which take account of a known characteristic of the human visual system, also known as visual saliency. In recent years, saliency-guided mesh simplification or polygon reduction methods, or surface saliency evaluated by simplification have been proposed. Since surface distinction identifies parts that are consistent within a class and distinct from other classes, we expect the simplification algorithm to preserve distinctive features better than other approaches: in other words, that the distinctive features of each model should be simplified less than the rest of the mesh. Saliency maps in themselves do not suffice to be used as the main simplification metric; rather they function as support data and heuristics for simplification.

The most popular simplification method used for saliency detection evaluation has focused on minimizing geometric errors: frequently, the Quadric Error Metric(QEM) proposed by Garland and Heckbert [1]. In particular, Howlett et al. [97] used eye fixation equipment to capture the salient regions, and then to evaluate them by applying QEM-based simplification.

For evaluating the effectiveness of the mesh saliency method proposed by Lee et al. [3], the authors have modified the quadrics-based simplification method (Qslim) of Garland

and Heckbert by weighting the quadrics with mesh saliency. Shilane et al. [40] follow Lee's work by weighting vertices instead with mesh distinction scores.

Bulbul et al. [49] attempted to delay the simplification of the salient parts of the mesh, since those parts are presumably the parts of the mesh that viewers focus on. Tan and Chau [98] proposed a saliency-guided image-driven simplification method. They enhance the edge collapse error metric by weighting it with saliency, and delay edge collapses in salient regions by artificially increasing their error. Williams [48] used a simple perceptually-based image distortion metric to guide view-dependent simplification. Qu et al. [47] observed that the fact that textures and lighting can visually mask mesh distortion could also be exploited to guide simplification.

Figure 6.1 shows examples of such saliency-guided simplification results. It may be seen that more vertices are preserved in the salient regions, and the simplified meshes faithfully represent both the global shape and the local geometry.

6.1.1 Test Data

In order to show the robustness of our proposed methods, two range image datasets were used in our project, which were downloaded from two publicly-available range image database hosted by the Signal Analysis and Machine Perception Laboratory at Ohio State University (OSU)(`http://sampl.eng.ohio-state.edu/sampl/database.htm.`), and the Stuttgart Range Image Database (`http://range.informatik.uni-stuttgart.de/htdocs/html/`), respectively.

A. OSU Range Image Database

Examples from the OSU range image dataset are shown in Figure 6.2. All of these images were captured using a Minolta Vivid 700 range camera with a resolution of 200 by 200 pixels, and were subject to a camera motion with rotation angle of 10, 20, or 36 degrees, respectively, around an unknown rotation axis followed by a translation in 3D space. From this dataset, we selected several models and reconstructed them as 3D models: the results may be seen in Figure 6.3.

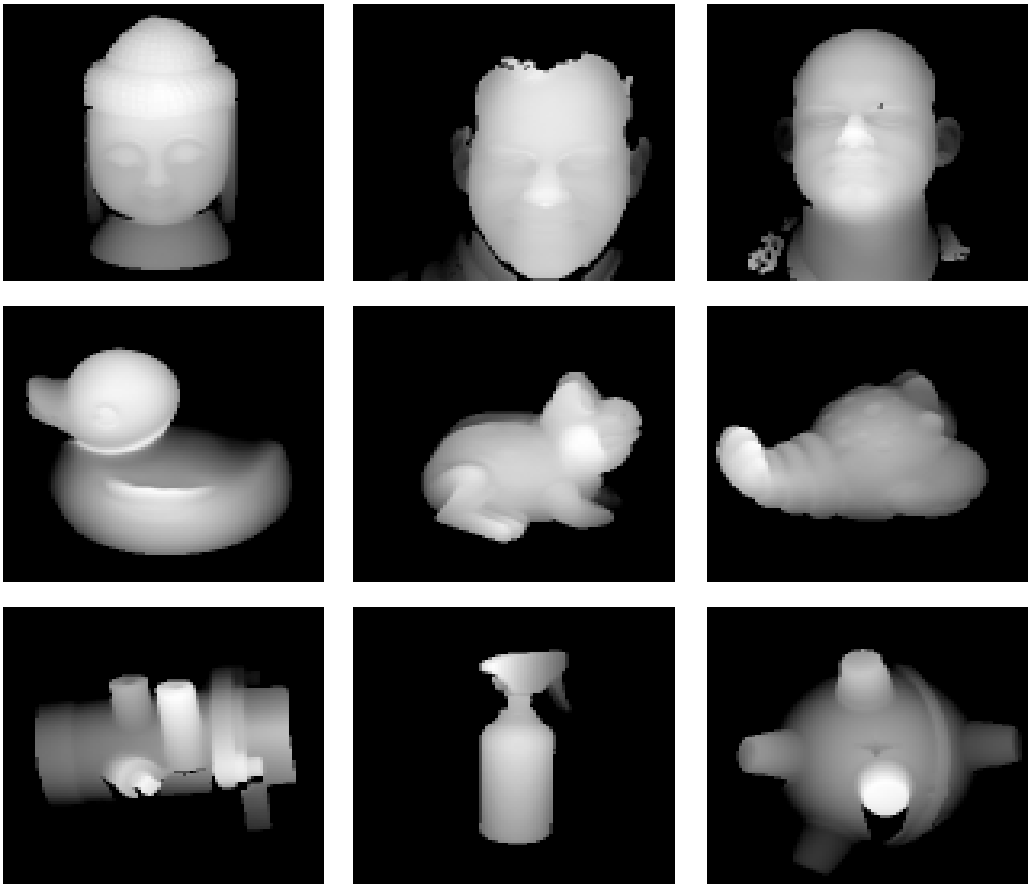


FIGURE 6.2: Samples of OSU real range images. From top left to bottom right: *buddha*, *pat*, *rick*, *duck*, *frog*, *lobster*, *valve*, *bottle* and *frame*.

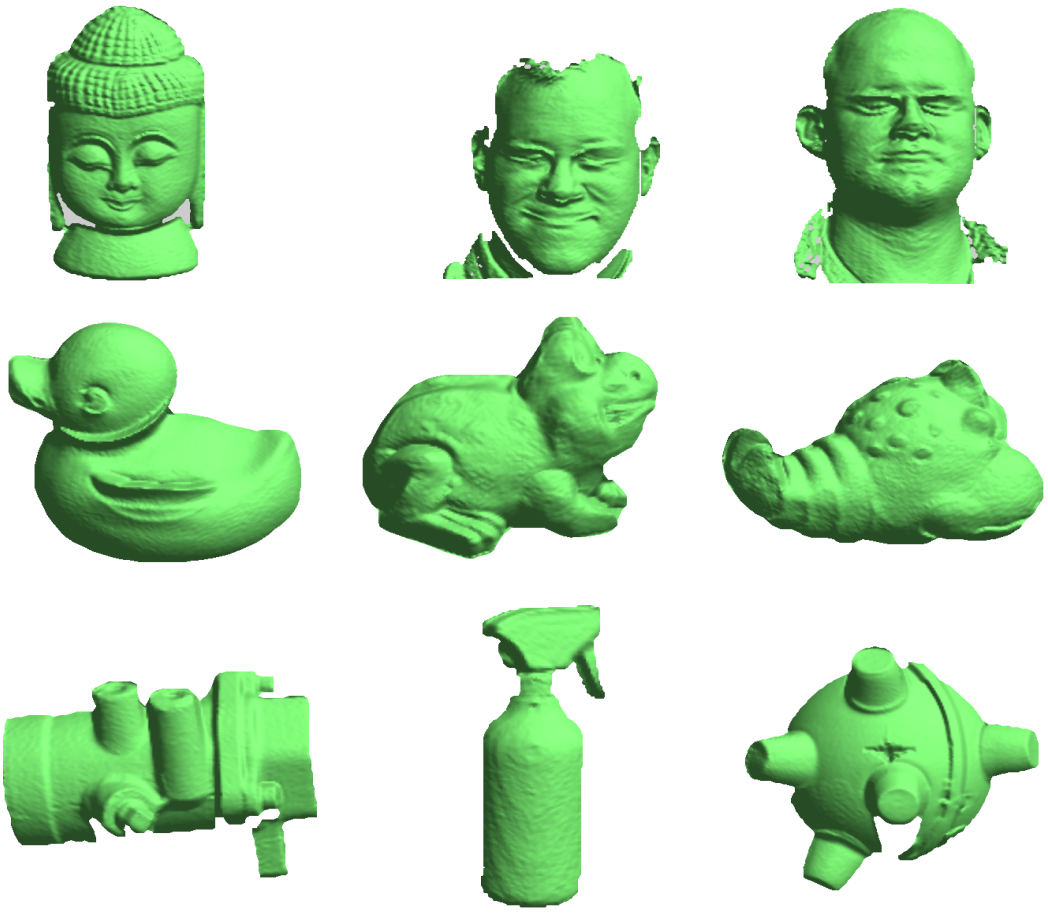


FIGURE 6.3: Reconstructed 3D models from the OSU range images. From top left to bottom right: *buddha*, *pat*, *rick*, *duck*, *frog*, *lobster*, *valve*, *bottle* and *frame*.

B. Stuttgart Range Image Database

In order to show the robustness of our saliency detection methods, we used two data sets in this chapter, the second being taken from the Stuttgart Range Image Database (SRID) - a collection of synthetic range images taken from high-resolution polygonal models available online. Compared with the dataset from OSU, the SRID range images have higher resolution, at 400*400, and each model is imaged from more viewpoints: 258. Examples of images from the Stuttgart Range Image Database are shown in Figure 6.4. The reconstructed 3D models from these range images are shown in Figure 6.5.

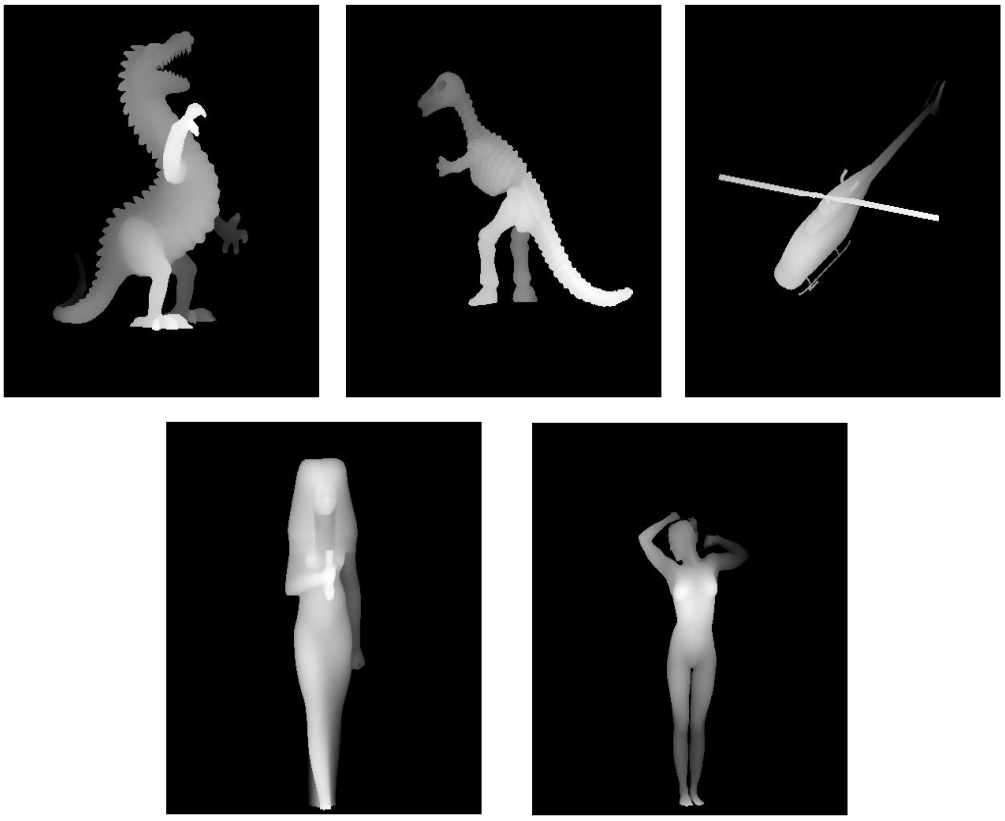


FIGURE 6.4: Samples of Stuttgart range images used. From top left to bottom right: *dragon*, *dinosaur*, *copter*, *isis*, and *female*.

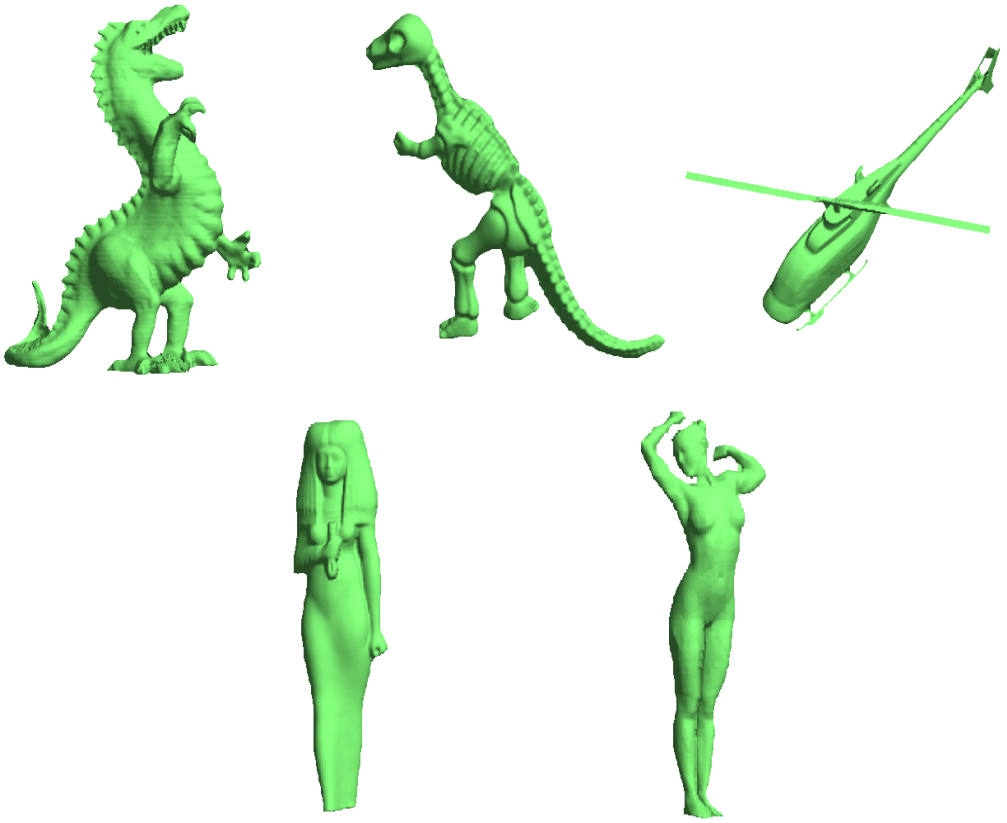


FIGURE 6.5: Reconstructed 3D models of the Stuttgart range images. From top left to bottom right: *dragon*, *dinosaur*, *copter*, *isis*, and *female*.

6.2 Simplification Criteria

Most existing saliency-guided simplification methods [3, 40, 47, 49] use the simplification approach proposed by Garland and Heckbert [1]. In particular, Lee et al. used their mesh saliency to weight vertices in a quadrics-based simplification scheme. In this section, we use the same simplification approach.

To review: quadric error simplification works by contracting the edge that has the least quadric error. The quadric error for each vertex is a measure of how far that vertex has moved during simplification. Let P be the set of triangular faces incident at a vertex v, where the triangular face $p \in P$ is defined by the equation $ax + by + cz + d = 0$, $a^2 + b^2 + c^2 = 1$, is represented as $(a, b, c, d)^T$. The quadric for the face p is then defined as $Q_p = pp^T$. The quadric error E of v is the sum of all the quadrics of neighboring planes:

$$E_v = v^T(\sum_{p \in P} Q_p)v. \tag{6.1}$$

Based on the saliency value $Saliency\_final$ for each vertex, the definition of the new quadric error for every edge e is defined as

$$E_e = (Saliency\_final)_{v_i} \cdot E_{v_i} + (Saliency\_final)_{v_j} \cdot E_{v_j}. \tag{6.2}$$

As mentioned previously, when we delay the simplification of the salient parts of the surface, strengthen the simplification of non-salient regions, we have found that using the simplification weights requires a non-linear amplification of the saliency: the high-saliency vertices are preserved longer than the non-salient vertices. To accentuate the difference between salient and non-salient vertices, we set the lower 65% of vertices to the minimal saliency value (non-salient). After each edge is collapsed, the new vertex is assigned an error that is the maximum of the two vertices collapsed, so that salient vertices preserve their scores without being averaged with nearby areas.

6.3 Comparison Methods

As we outlined above, there are several saliency detection methods that have been applied to simplification [3, 40, 47, 49]. In this section, the saliency detection methods evaluated in our experiments include: Mesh Saliency (MESA) and Distinctive Region (DIRE). In the interests of fair comparison, methods have been re-implemented in MATLAB, although the original source code for MESA is in C++, and is available online to be

downloaded. A quantitative comparison between our method and these two competing saliency-guided simplification techniques is provided in the following section.

6.3.1 Mesh Saliency (MESA)

Lee et al. [3] introduced the idea of mesh saliency as a measure of regional importance for graphics meshes, and explored the applications of mesh saliency to mesh simplification and view selection. The mesh saliency was computed by using a center-surround mechanism, as the center-surround mechanism has the intuitive appeal of being able to identify regions that are different from their surrounding context. The basic idea is to filter the curvatures of vertices in meshes using a center-surround operator on Gaussian-weighted mean curvatures. The mesh saliency algorithm can be summarized as follows:

1. Compute the curvature of each vertex $v \in M$ by using Taubin's method. The $k_H(v)$ indicates the mean curvature of v.

2. Let the neighborhood $N(v, \sigma)$ for a vertex, v, be the set of points within a distance σ, and x is a mesh point. (The Euclidean distance had been employed.) Then the Gaussian-weighted average of the mean curvature can be computed as:

$$G(k_H(v), \sigma) = \frac{\sum\limits_{x \in N(v, 2\sigma)} k_H(x) exp[-\|x - v\|^2/(2\sigma^2)]}{\sum\limits_{x \in N(v, 2\sigma)} exp[-\|x - v\|^2/(2\sigma^2)]} \tag{6.3}$$

   For the Gaussian filter, the Euclidean distance was assumed to be 2σ.

3. The saliency $\mathscr{S}(v)$ of a vertex v is the absolute difference of the Gaussian-weighted average between fine and coarse scale:

$$\mathscr{S}(v) = |G(k_H(v), \sigma_i) - G(k_H(v), 2\sigma_i| \tag{6.4}$$

   where σ_i is the standard deviation of the Gaussian filter at scale i.

4. Use five scales $\sigma_i \in \{2\varepsilon, 3\varepsilon, 4\varepsilon, 5\varepsilon, 6\varepsilon\}$, where ε is 0.3% of the length of the diagonal of the bounding box of the model. The final mesh saliency is then computed by adding the saliency maps at each of the five scales after applying a non-linear normalization of suppression.

6.3.2 Distinctive Region (DIRE)

Shilane and Funkhouser [40] analyzed distinctive regions by performing a shape-based search using each region as a query into a database. The distinctive regions have shape

consistent with objects of the same type, and different from objects of other types. The main steps of this approach are as follows:

1. The first step is to define a set of local regions covering the surface of the object. The authors generate for each mesh a set of random points that are the centers of spherical regions covering its surface at multiple scales. The sampling method is a modified version of Osada's work [270]: a number of points are allocated to every triangle in proportion to its surface area, and the allocated number of points is then sampled from every triangle uniformly with respect to its surface area.

2. In the second step of this technique, the shape of every spherical region is described by means of a Harmonic Shape Descriptor (HSD) with 32 spherical shells and 16 harmonic frequencies, in which the shape descriptor offers rapid computation, is compact in storage, is concise to store, indexable and easy to compare. The HSD is computed for all regions of a single surface by using a 3D grid containing a Gaussian function of the surface's Euclidean Distance Transform (GEDT). A spherical grid is then constructed by computing the GEDT at regular intervals of radius and polar angles for every spherical region centered on a point sampled from the surface.

3. The third step is the identification of the distinctive regions. The authors compute the degree of distinctiveness of every shape descriptor with respect to a database containing multiple classes of objects. The distance of any one of the features in the given model to the closest descriptor is given, and the distinctiveness of the descriptor is then computed by evaluating a retrieval performance metric: Discounted Cumulative Gain (DCG).

6.4 Measurement Criteria

There are several mesh error measurements. In this section, we performed a quantitative comparison between our method and other comparative methods by measuring the Root Mean Square Error (RMSE) and Metro errors [253] between the original and the simplified mesh, in the expectation that these measures will reflect the visual and geometrical differences between them. RMSE is used as an efficient measurement in the mesh simplification. The frequently-cited Metro evaluation method is a tool designed to evaluate the difference between two triangular meshes.

Metro adopts an approximation approach based on surface sampling and point-to-surface distance computation. It evaluates the distance between meshes, and works efficiently. The Metro Error between the original mesh M and the simplified mesh M' is such that:

$$Metro_{(M,M')} = \sqrt{\frac{1}{A}\int dist(p,M)^2 ds} \tag{6.5}$$

where $dist(p, M)$ is the distance between the new vertex p in M' and a triangle of the original mesh M that is closest to p. A is the sum of the areas of all triangles of the mesh in its simplified version.

Although neither of these tools measures how well salient areas are preserved, they are useful tools when comparing the results of two different ways of simplifying meshes based on saliency. We can see to what extent preserving saliency has had a deleterious effect on faithfulness to the original mesh. Measurement of these errors then discloses the effectiveness of the detected saliency when using different methods.

6.5 Experimental Results

For comparison's sake, in this section we gather the simplification results of methods MESA, DIRE, VBS, and RBS, respectively. The comparison is made not only from the the visible results of the simplification process on objects derived from the two datasets, as shown in Figure 6.6 to Figure 6.9, and Figure 6.11 to Figure 6.14, respectively, but also through the error measurements (Figure 6.10 and Figure 6.15). All the results are obtained using a 95% simplification rate, which means that 95% of the original vertices were removed. It may be observed that the simplification results of most of these methods retain the major topological characteristics of the initial models.

In the case of MESA, it is hard to distinguish the local features of the model *buddha* in Figure 6.6: the curly hair is entirely misrepresented, and the mouth region is transformed

almost into a flat region. Figure 6.7 shows the model *lobster*. None of the prominent bumpy area has been preserved by MESA. These results show that pure mean curvature as a measure of saliency detection and the QEM from points to faces can hardly capture the salient regions.

Looking at the results for the SRID models, the eye socket of model *dinosaur* is retained in Figure 6.11, but is reconstructed with large distortions. The face of model *female* is totally over-smoothed in Figure 6.13, as is that of the *isis* model in Figure 6.14.

In summary, the MESA gives poor results under a high simplification rate. It is struggles to reconstruct a simplified model that is pleasing to human visual perception, since this method generates relative poor saliency results when compared with the competing DIRE, VBS and RBS methods.

The performance of DIRE is better than that of MESA, as this method use quadric error metrics but also incorporates a better saliency detection method, and the detected saliency as a supporting decision factor. As may be seen in Figure 6.6, most of the facial features of model *buddha* have been retained. However, from the snapshot of Figure 6.6, we note that distortion has occurred in the eye regions. In Figure 6.8, the pipe orifice is blurred: it is difficult to describe the shape of the orifice. DIRE has preserved most of the features of model *buddha* in Figure 6.6: however, distortion has ruined the topology of the eyes and eyebrows.

For the SRID models, the teeth of model *dragon* in Figure 6.12 are blurred: the DIRE algorithm misdetects in this region. For the same reason, the face of model *isis* in Figure 6.14 is also over-smoothed: only a few points are detected as salient by the DIRE method.

As we mentioned above, DIRE-guided simplification is an extension from the MESA technique. It incorporates the mesh saliency with quadric error metrics. However, MESA does not detect the whole eye regions as salient: vertices in the eye region are partially preserved, and partially missed.

By contrast, our proposed methods, VBS and RBS, achieved visually pleasing results. Both not only retained the model's global shape, but also preserved significant local features, such as the curly hair, eye, nose and mouth of *buddha*; the bumpy area of *lobster*; the pipe orifice of *valve*; and the facial detail of *rick*, *female*, and *isis*. The resulting models have the highest surface fidelity of any of the competing methods, since the proposed methods take into account two aspects of information in simplifying the models: local, and global. On careful observation of the results of VBS and RBS, we find that RBS gives better performance than VBS. As we described in the last chapter, the way in which RBS detects salient regions not only involves the vertices, but also

the regions, which means that both the local and global information are considered, providing a more faithful saliency result. Depending on the simplification criteria, more vertices are sampled in the salient regions.

The outstanding performances of the proposed methods are confirmed in Figure 6.10 and Figure 6.15. All the models were simplified with three different simplification rates: 50%, 80%, and 95% respectively. The simplification results were illustrated by the RMSE and Metro errors. As the simplification rate increases, the RMSE and Metro errors become much larger for all of the simplification methods, as expected. On careful scrutiny of all cases, it may be seen that the errors of our proposed methods VBS and RBS achieve better simplification results than the alternatives in the parameter of RMSE or Metro error. For example, Figure 6.6 shows that in the case of the model *buddha*, the RMSE of MESA is 11.3×10^{-3} when the simplification rate is 50%, while the errors are reduced to 7.1×10^{-3} and 6.5×10^{-3} by VBS and RBS, respectively, which means that the errors have been reduced by as much as 26.8% and 33%. Figure 6.12 illustrates that the accuracy of VBS is 46.5% higher than MESA in model *valve* with a 95% simplification rate, while RBS is 56.3% more accurate than MESA.

On the SRID models, as shown in Figure 6.15, in the case of model *dinosaur*, when the simplification rate is 95%, the RMSE errors of VBS and RBS are 22.2% and 29.6% lower than that of DIRE. Taking into account the simplification errors presented in Figure 6.10 and Figure 6.15, it may be concluded that our methods achieve better simplification results than the best comparable existing methods.

In summary, all results show that the proposed method can not merely preserve small geometrical features much more faithfully than other simplification methods, but that it also consistently achieves significantly lower simplification error rates.

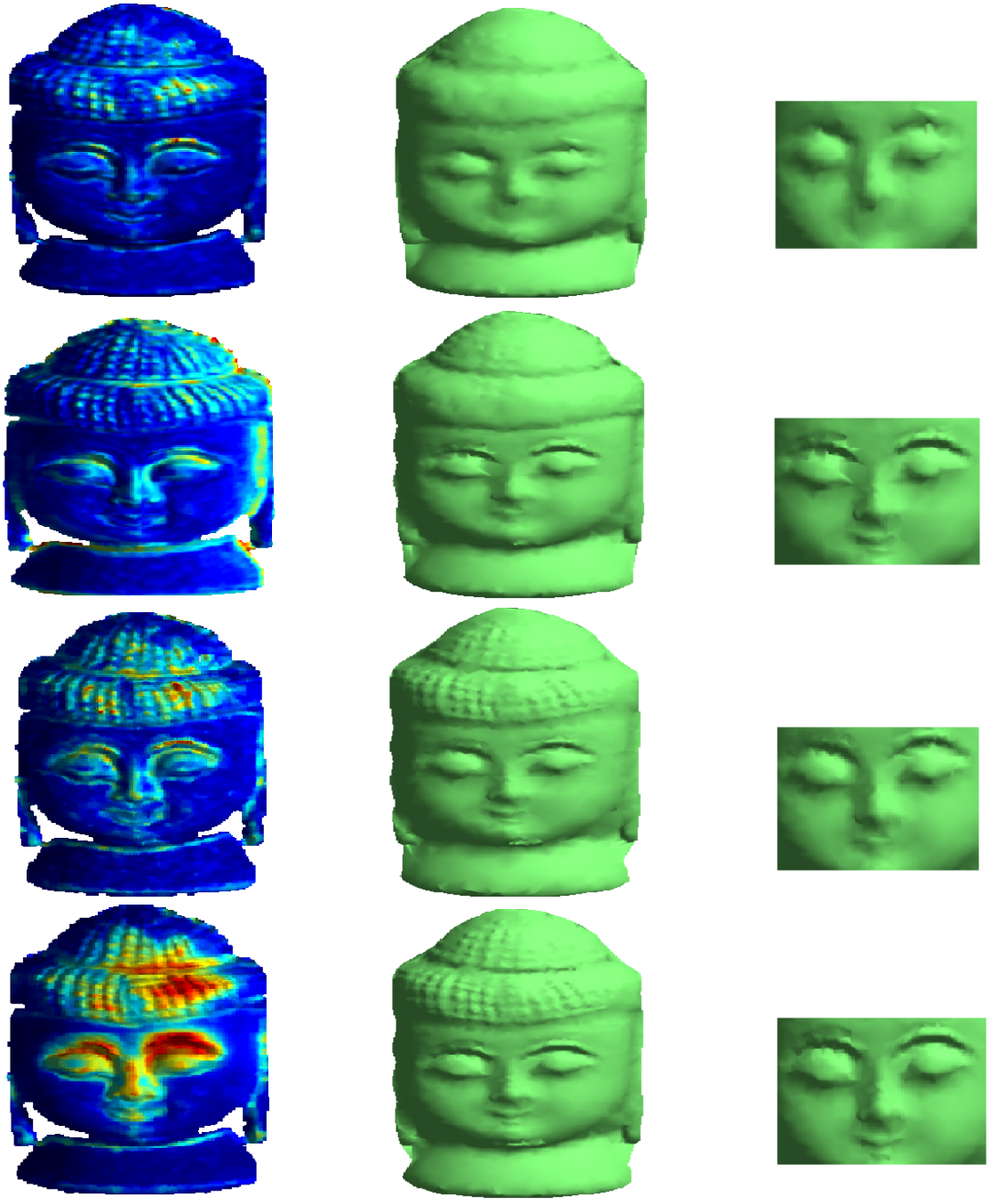


FIGURE 6.6: Examples of simplification results obtained by different methods on model *budda* from the OSU database, with 95% simplification rate. The snapshots illustrate the local details after simplification. The number of original vertices is 14866, and the simplified model contains 740 vertices. First row: MESA-guided simplification result. Second row: DIRE-guided simplification result. Third row: VBS-guided simplification results. Last row: RBS-guided simplification results.

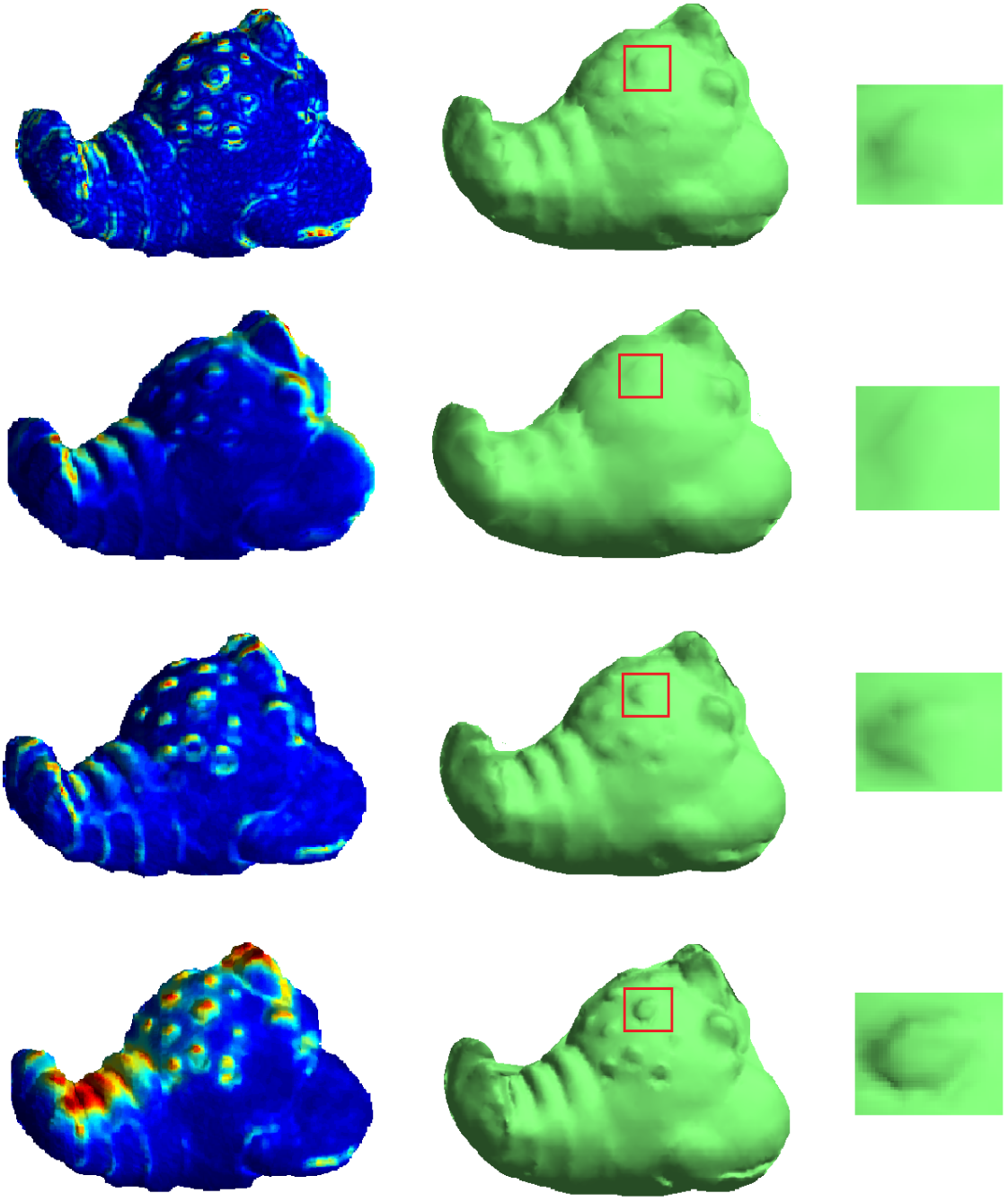


FIGURE 6.7: Examples of simplification results obtained by different methods on model *lobster* from the OSU database, with 95% simplification rate: the snapshots illustrate the local details after simplification. The number of original vertices is 11759, and the and simplified model contains 588 vertices. First row: MESA-guided simplification result. Second row: DIRE-guided simplification result. Third row: VBS-guided simplification result. Last row: RBS-guided simplification result.

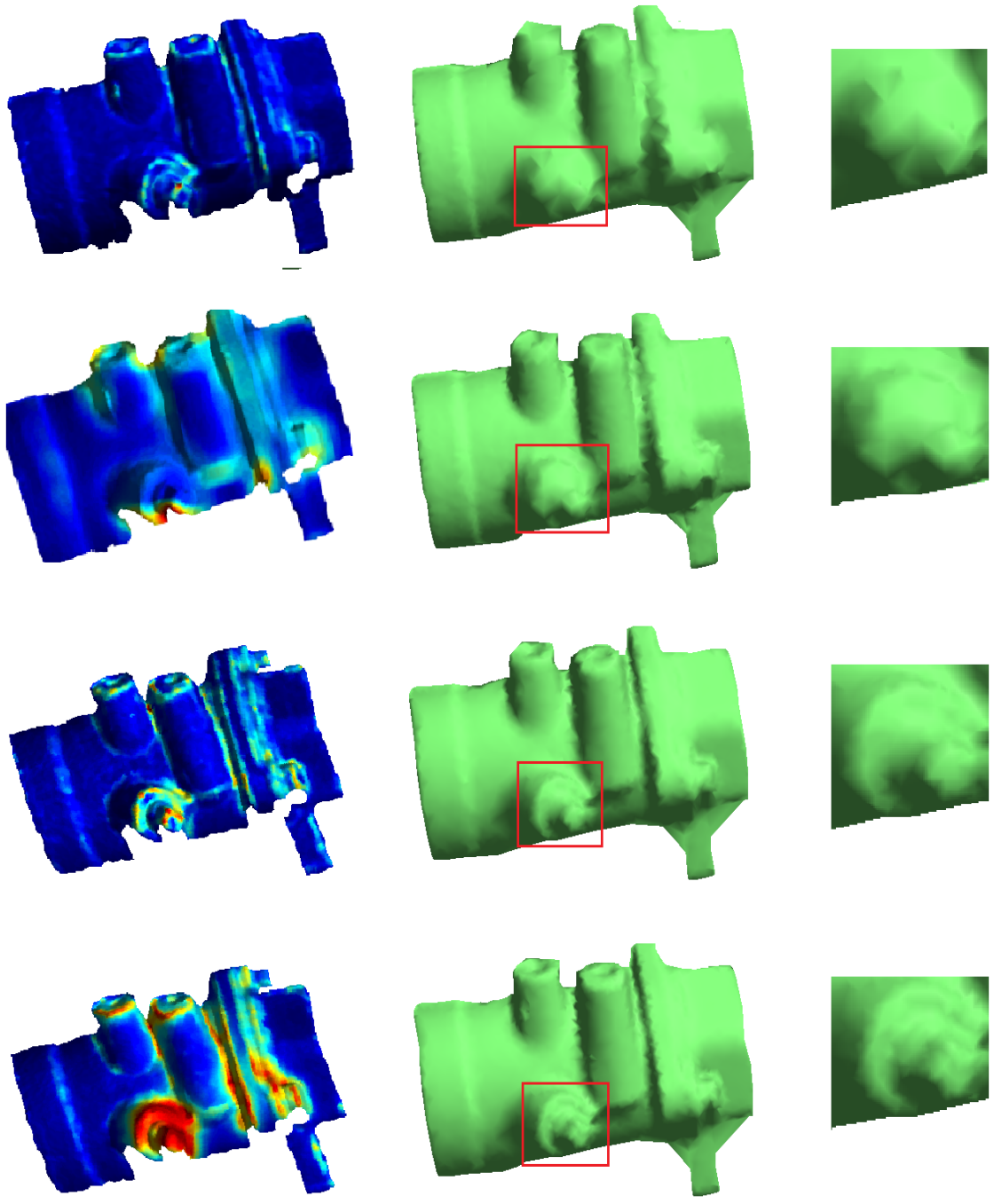


FIGURE 6.8: Examples of simplification results obtained by different methods on model *valve* from the OSU database, with 95% simplification rate: the snapshots illustrate the local details after simplification. The number of original vertices is 12787, and the simplified model contains 640 vertices. First row: MESA-guided simplification result. Second row: DIRE-guided simplification result. Third row: VBS-guided simplification result. Last row: RBS-guided simplification result.

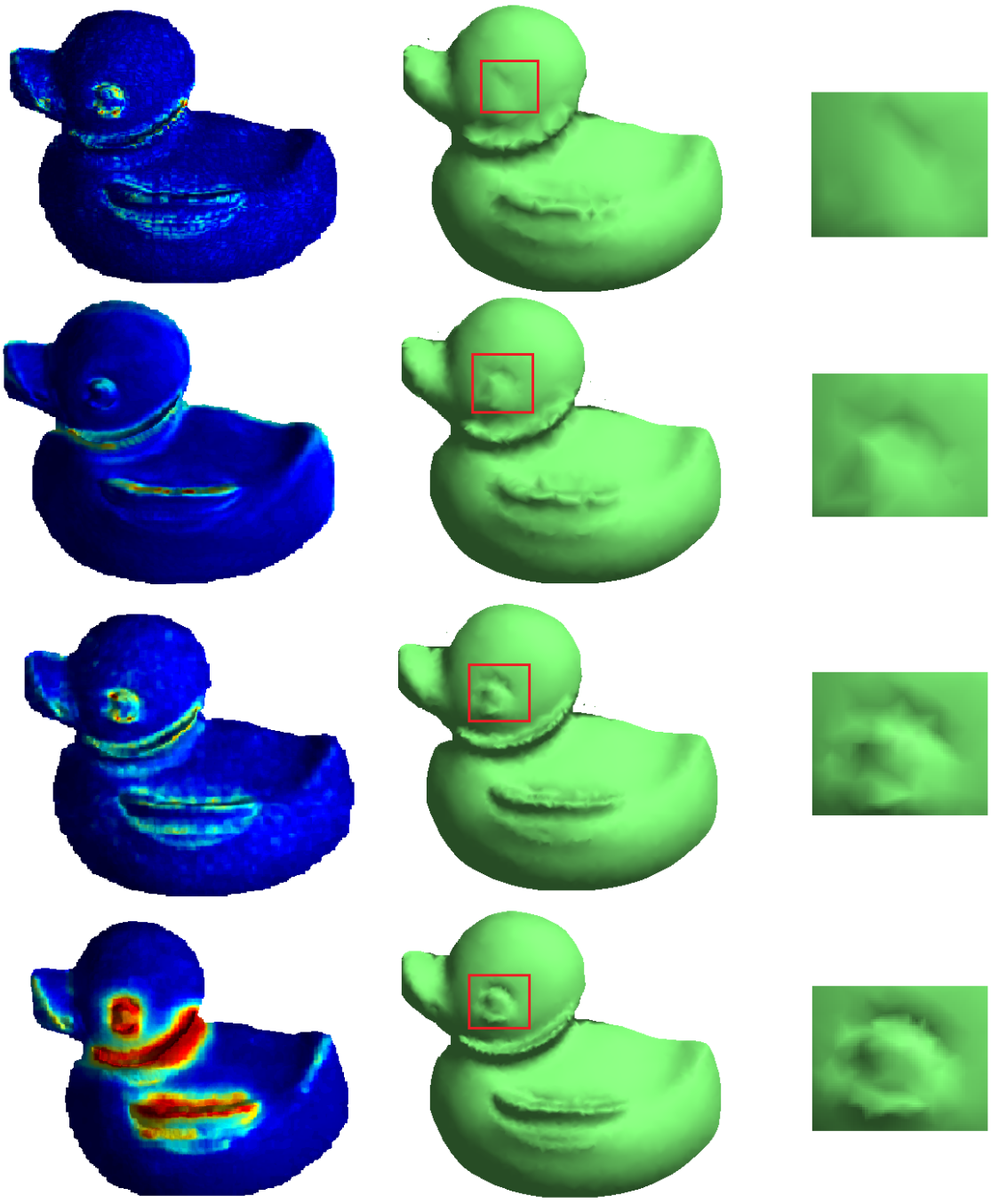


FIGURE 6.9: Examples of simplification results obtained by different methods on model *duck* from OSU database, with 95% simplification rate: the snapshots illustrate the local details after simplification. The number of Original vertices is 15899, and the simplified model contains 795 vertices. First row: MESA-guided simplification result. Second row: DIRE-guided simplification result. Third row: VBS-guided simplification result. Last row: RBS-guided simplification result.

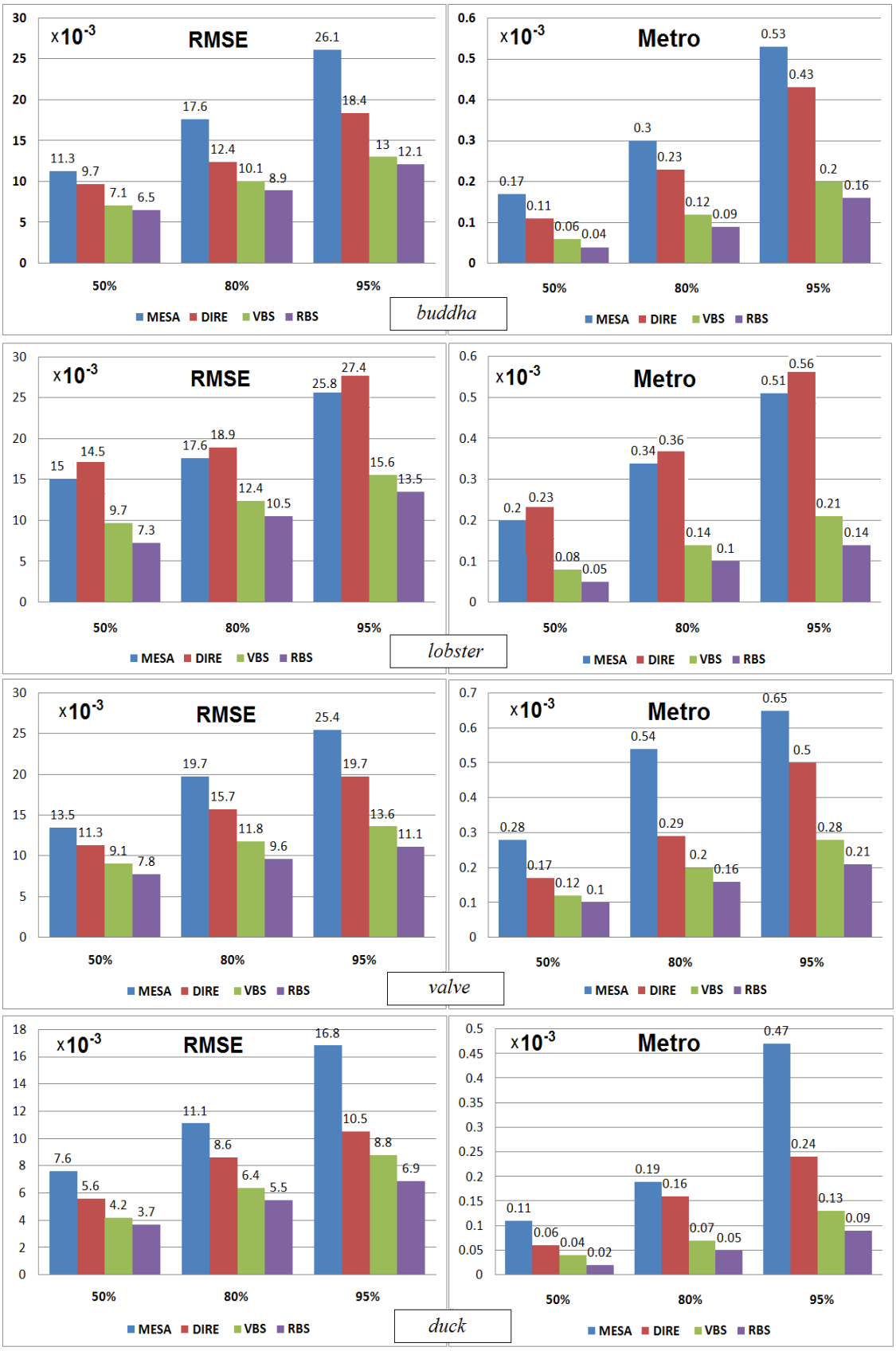


FIGURE 6.10: RMSE and Metro errors measured with simplification rates of 50%, 80%, 95% by different simplification methods: MESA, DIRE, VBS, and RBS on (top to bottom) models *buddha*, *lobster*, *valve* and *duck*.

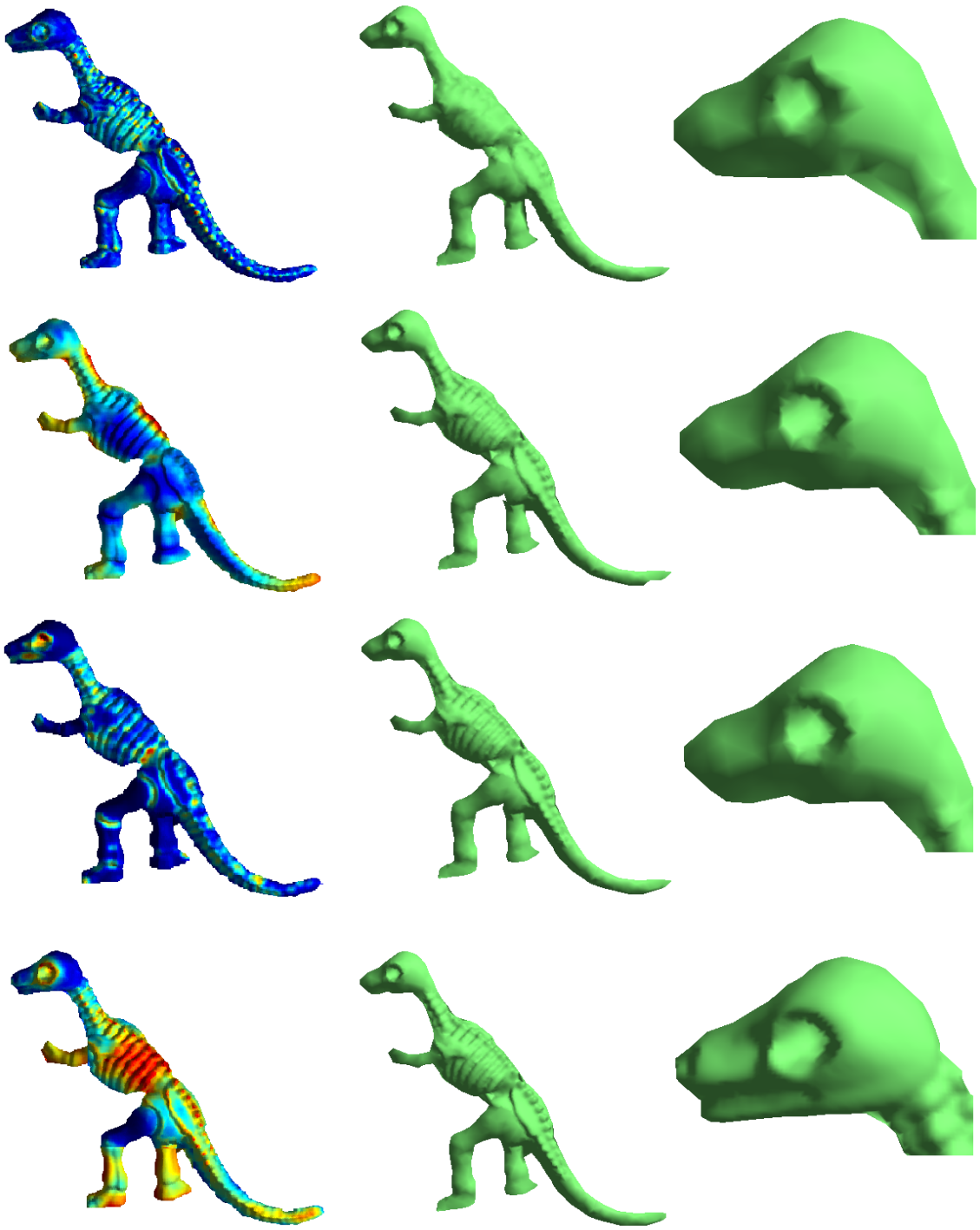


FIGURE 6.11: Examples of simplification results obtained by different methods on model *dinosaur* from the SRID database, with 95% simplification rate: the snapshots illustrate the local details after simplification. The number of original vertices is 15160, and the simplified model contains 758 vertices. First row: MESA-guided simplification result. Second row: DIRE-guided simplification result. Third row: VBS-guided simplification result. Last row: RBS-guided simplification result.

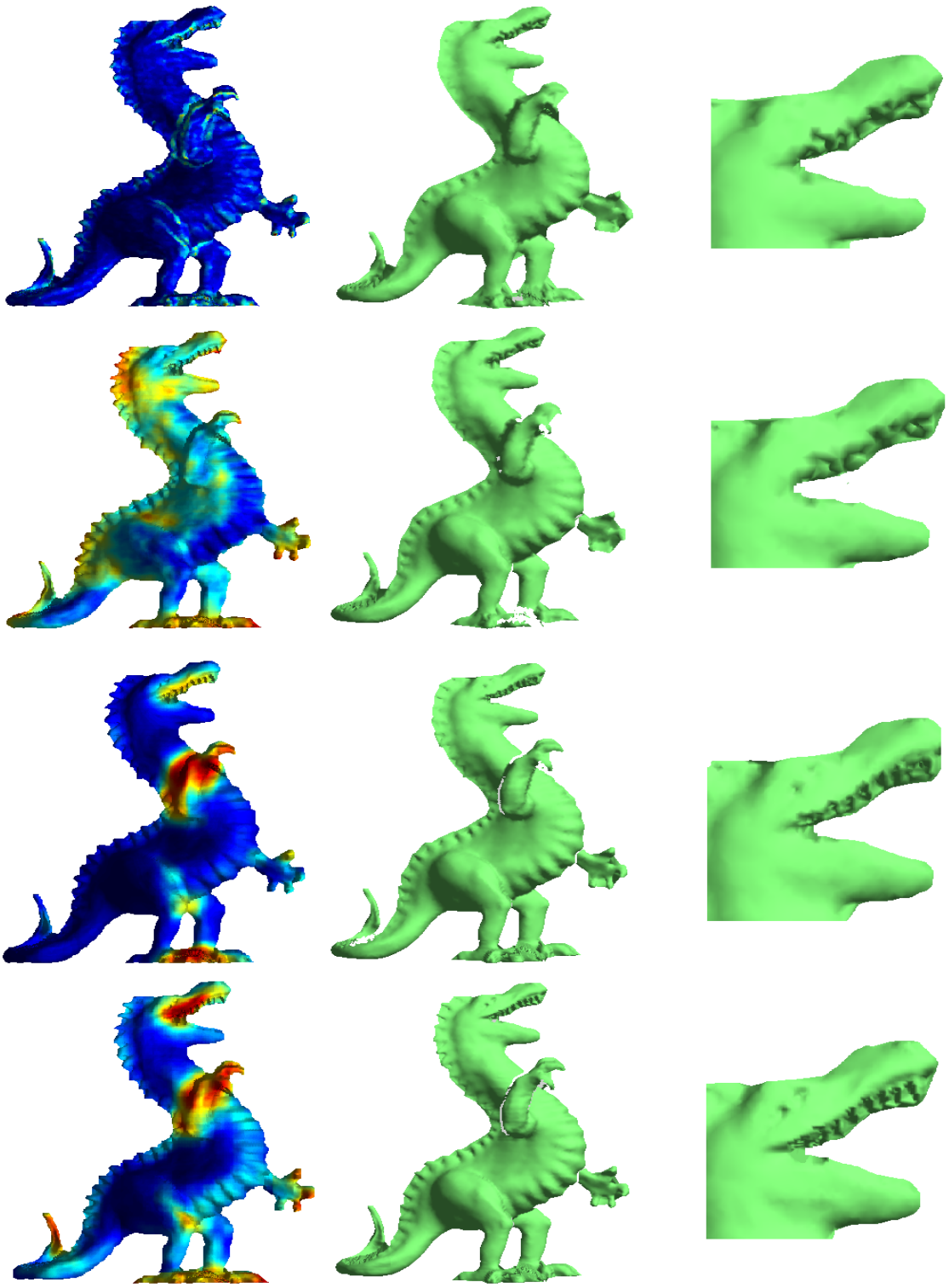


FIGURE 6.12: Examples of simplification results obtained by different methods on model *dragon* from the SRID database, with 95% simplification rate: the snapshots illustrate the local details after simplification. The number of original vertices is 28730, and the simplified model contains 1436 vertices. First row: MESA-guided simplification result. Second row: DIRE-guided simplification result. Third row: VBS-guided simplification result. Last row: RBS-guided simplification result.

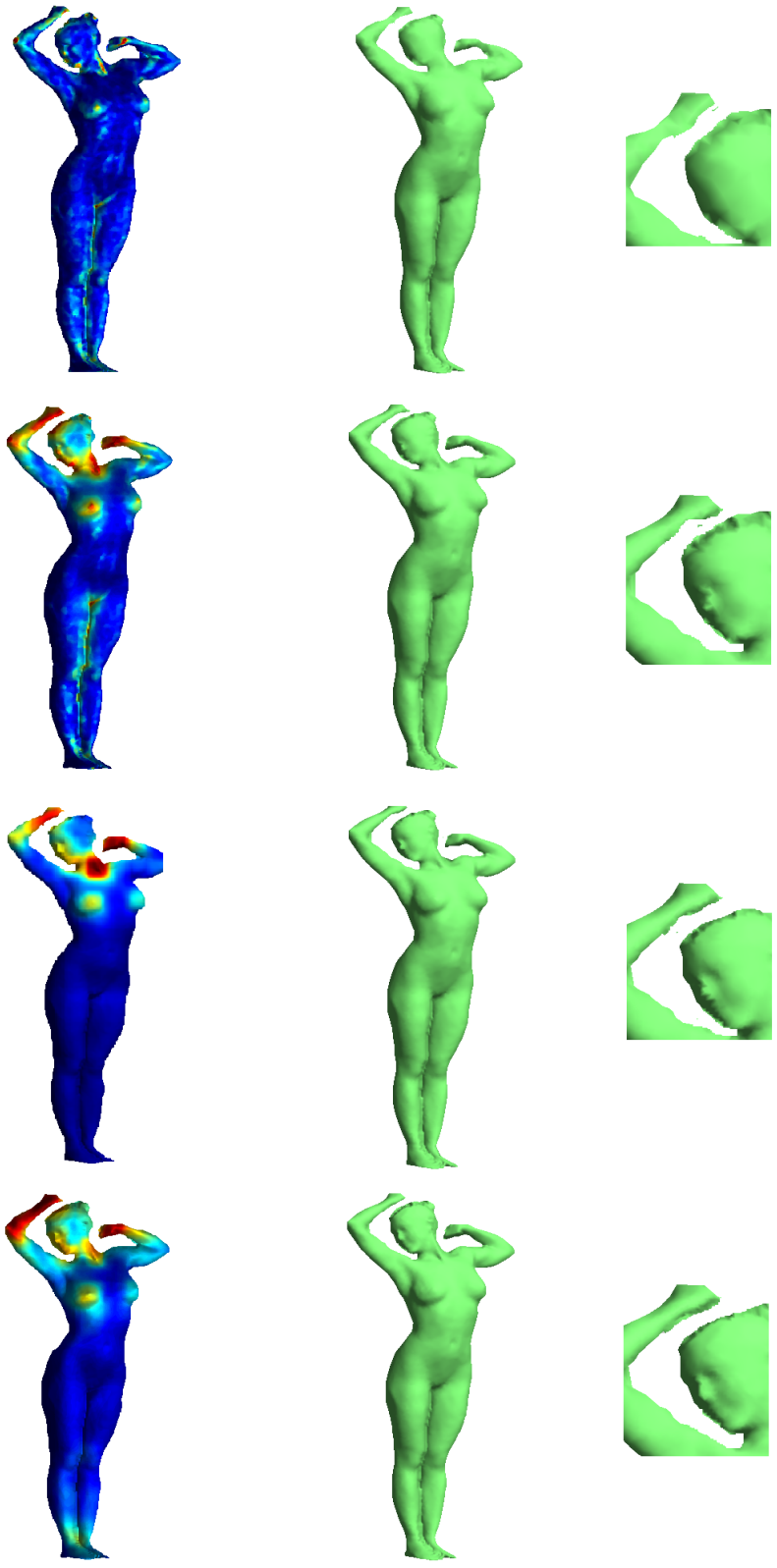


FIGURE 6.13: Examples of simplification results obtained by different methods on model *female* from the SRID database, with 95% simplification rate: the snapshots illustrate the local details after simplification. The number of original vertices is 14781, and the simplified model contains 739 vertices. First row: MESA-guided simplification result. Second row: DIRE-guided simplification result. Third row: VBS-guided simplification result. Last row: RBS-guided simplification result.

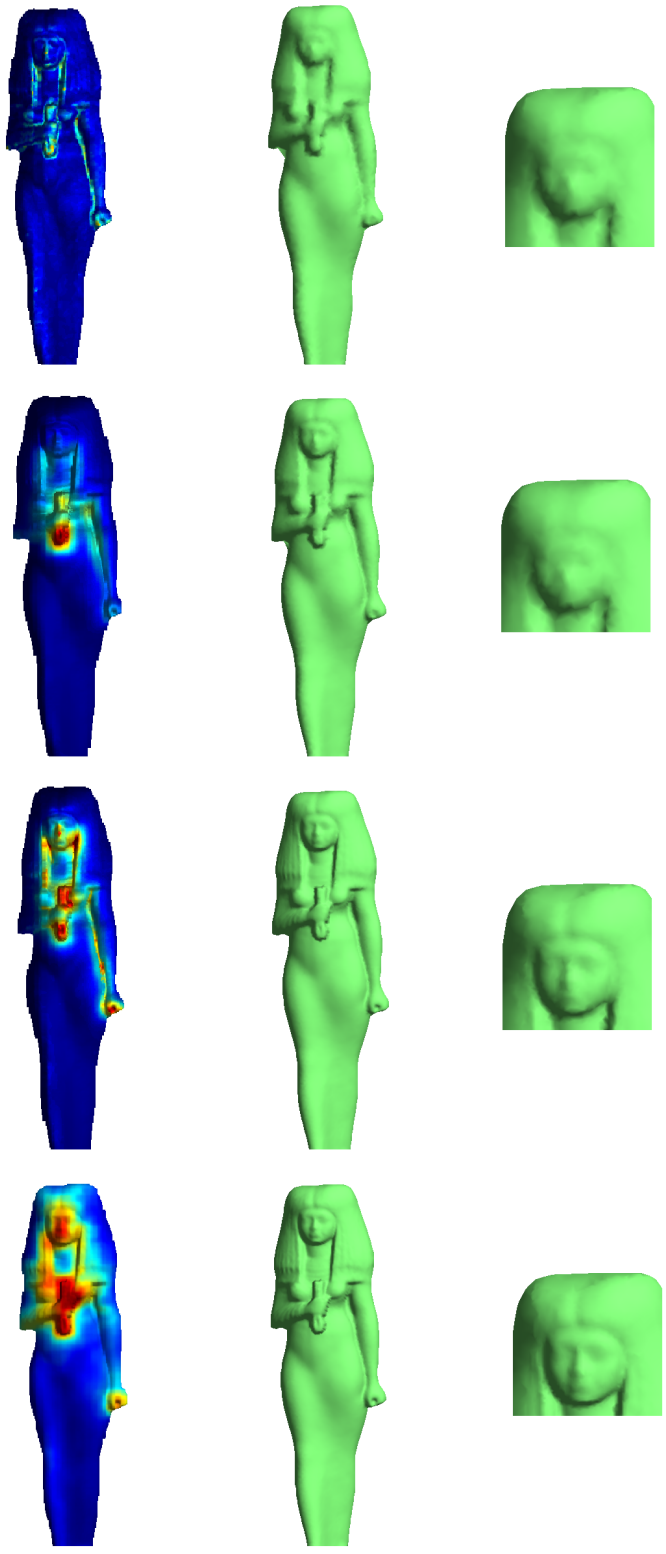


FIGURE 6.14: Examples of simplification results obtained by different methods on model *isis* from the SRID database, with 95% simplification rate: the snapshots illustrate the local details after simplification. the number of original vertices is 25702, and the simplified model contains 1285 vertices. First row: MESA-guided simplification result. Second row: DIRE-guided simplification result. Third row: VBS-guided simplification results. Last row: RBS-guided simplification results.

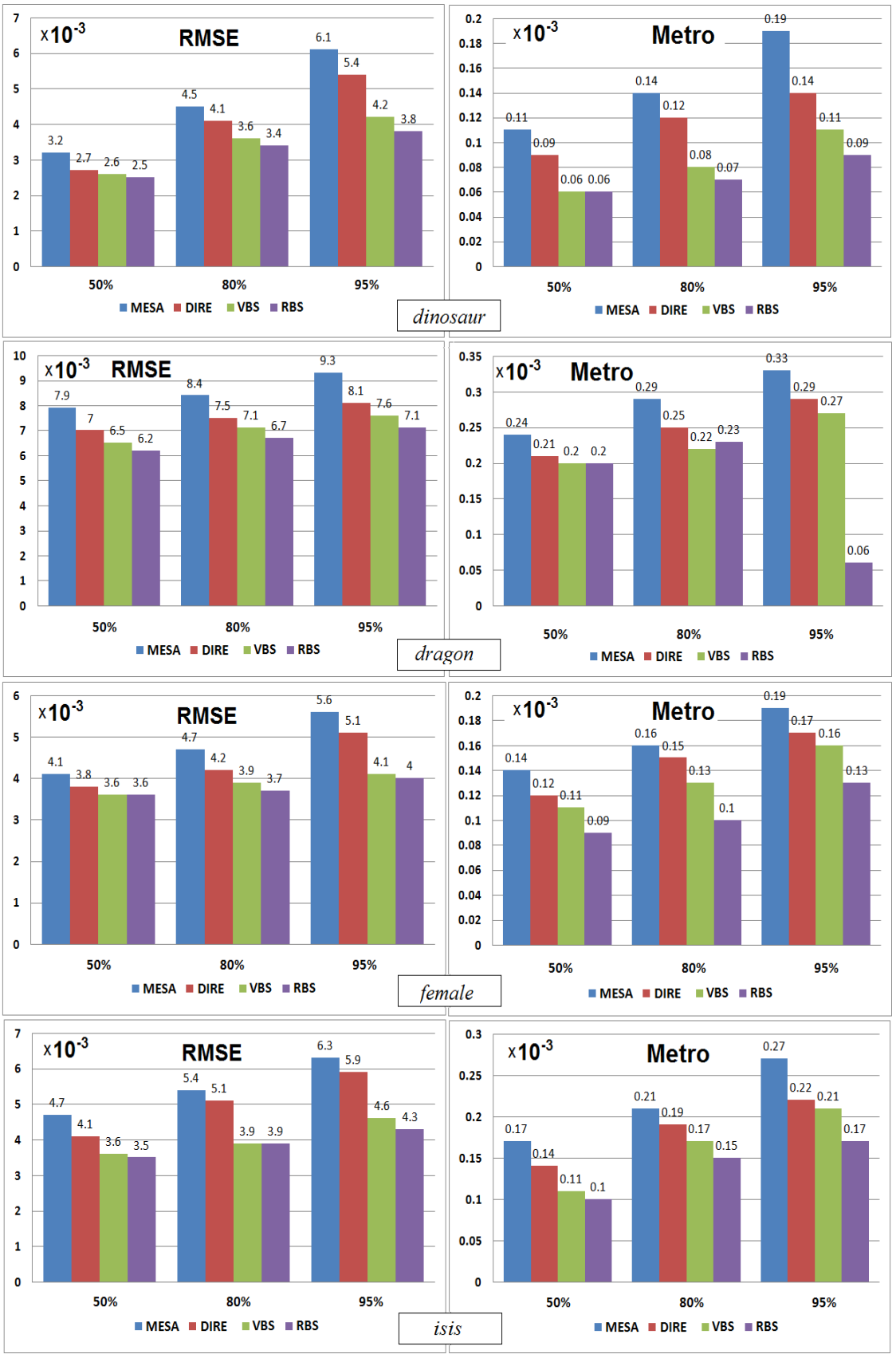


FIGURE 6.15: RMSE and Metro errors measured with simplification rates of 50%, 80%, 95% by different simplification methods: MESA, DIRE, VBS, and RBS on (top to bottom) models *dinosaur*, *dragon*, *female* and *isis*.

6.6 Summary

We have demonstrated that detected salient regions can be incorporated into 3D image applications by focusing an algorithm on the salient regions of a surface. Our saliency-guided simplification algorithms automatically detect the salient regions, and the simplification was achieved by using weighted quadric error metrics. In addition, a non-linear amplification of the saliency is required to estimate the simplification weights: this preserves more vertices from the salient region, and fewer vertices from the non-salient region. For a number of examples, we have shown that the most salient/important regions are well preserved after simplification.

The superior performance of our method also has been validated by numerical measurement - mesh error. In this chapter, the RMSE and Metro error were employed, and the results showed that our methods produced better results: both of our methods achieved much lower mesh errors by RMSE and Metro measurements.

This application of salient region detection is first step. We believe that there is a wealth of new ways to utilize salient region detection in these and other applications in computer graphics and vision.

Chapter 7

Saliency-guided Interest Points Detection

In recent years, there has been a growing interest in the use of detailed models for better representation. Consequently, most of the latest 3D scanners can generate huge quantities of data points within a limited time. This creates a number of challenges for storage, editing and transmission. These challenges place a greater burden on feature detection tasks. The detection of interest points is an important step for the analysis of mesh surfaces. 3D interest points detection is helpful in capturing the property of a point or region on a surface.

In the last chapter, we used the detected saliency to guide simplification, and showed that the proposed saliency detection method sufficiently improves the simplification results. Following on from this, in this chapter we present an application of the proposed saliency detection methods to interest point detection. The effectiveness of an application will be illustrated by the repeatability rate of the interest points. Moreover, the application will be validated by the transformation of the input data. Results show that our methods are both qualitatively and quantitatively better than the earlier techniques.

7.1 Introduction

The interest point detection topic emerged in the computer vision community, which was interested in reducing the amount of content used in high-level vision tasks. In past years, the evaluation of point feature detectors in the matching, recognition or texture classification in the 2D domain has been proposed. Interest points detection has a wide range of applications in the fields of 3D computer vision and graphics, such as shape

matching and object recognition. Automatic shape matching and object recognition are of great importance in computer vision. Many techniques have been developed to solve these problems using some novel interest point detection methods: e.g., SIFT. Use of interest points has the advantage of providing local features that are semantically significant and also invariant to rotation, scaling, noise, deformation, and articulation. A qualified interest points detection method for 3D models must have the following essential properties: it must be invariant to affine transformations, and robust to noise.

Recent years have witnessed the rapid development of methods for interest points detection on 3D surfaces. Most of the current 3D interest point detection methods have been developed over the last decade, as have defined functions summarizing the geometrical content of localities on a 3D model in multiple scales, and selected local extrema of those functions as interest points: e.g., curvature, extrema of which are assumed to correspond to candidate interest points. We have reviewed such methods in Chapter 2.

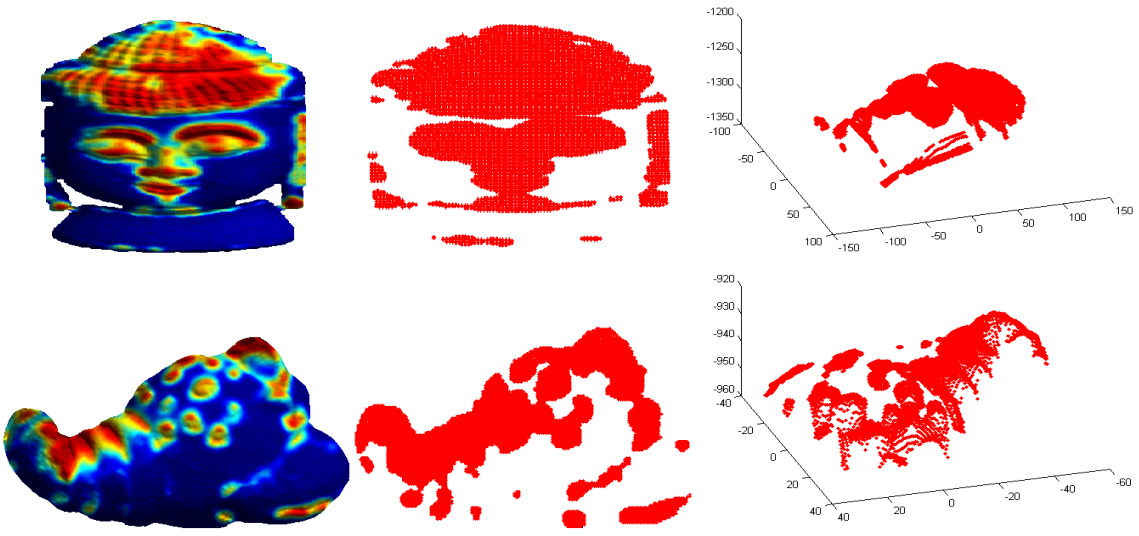
Some techniques were inspired by corresponding 2D approaches. However, such extensions are not always straightforward: for example, the keypoints represent interesting information at fine scales, and thus may be sensitive to noise and other transformations. Therefore, it is necessary to find larger interesting structures to overcome the problems at fine scales. In our work, we perform a comprehensive evaluation of our saliency-guided interest point detector, in which the interest points are sampled from the salient region by our proposed point sampling methods.

7.2 Interest Points Detection

A novel interest points detection method is proposed in this section. The interest points are selected by using the detected salient region. This method consists of three steps: salient region extraction, voxelization, and points selection.

7.2.1 Salient Region Extraction

The main idea of our interest points detection method is the detection of interest points mainly from the salient regions. Consequently, we need to extract the salient region first, which is easily implemented. In Figure 7.1(a), the warmer colors indicate the more-salient and the cooler colours the less-salient regions, with red as most salient and blue as least salient. The extraction of the salient regions may be demonstrated by setting a specific thresholding value. As we mentioned in Chapter 5, the salient values have been normalized into $[0, 1]$, and in our experiments, the vertices whose saliency



(a) (b) (c)

FIGURE 7.1: Example of the extraction of the salient region. (a) Saliency surface. (b) Extracted salient region. (c) Salient region seen from a different point of view.

values are larger than 0.5 will be extracted . Figure 7.1 shows the examples of extracted salient regions in models *buddha* and *lobster*.

7.2.2 Voxelization

In order to select points from the salient region with fairly even distribution, the voxelization is employed in our method. Voxelization is concerned with converting geometric objects from their continuous geometric representation into a set of voxels. This concept was first introduced by Arie Kaufman [271], and it has been explored broadly in many applications [272, 273].

Figure 7.2 illustrates an example of the voxelization of a triangle T. P shows the output of the voxelization of the triangle surface.

Let the model be voxelized by the bounding box, and the model may be divided into several sub-boxes. The resolution of the voxelization is specified by the user. A voxel model is a bounded 3D grid, so the first step is to decide its basic characteristics. Normally, the size of each block and the dimensions of the grid need to be decided. In this paper, we need enough resolution to make sure that important details are not lost. The dimensions of the grid have to be big enough to fit the entire model, but no bigger (to avoid inefficiency). This is fairly simple: we set the dimensions equal to the size of the bounding box of the surface triangles (the smallest box that encloses every

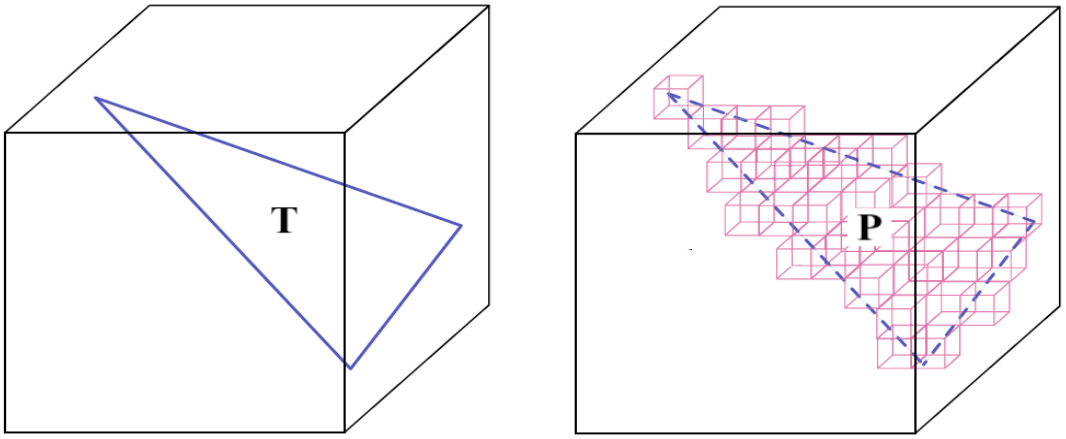


FIGURE 7.2: Example of voxelization. T is a triangular surface, and P is the voxelized surface.

point), rounded up to the nearest centimetre. The resolution of the bounding box can be controlled by the user in order to achieve a more accurate result for the point sampling.

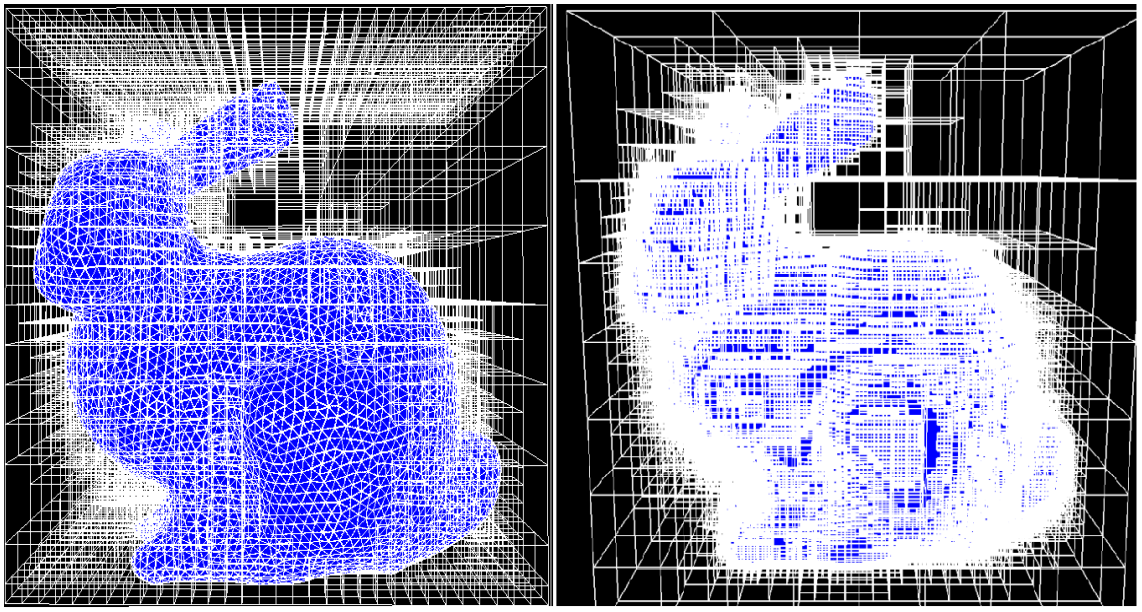
For example, Figure 7.3 shows the model *bunny* divided into voxels with different resolutions. The larger the number of the voxels is, the more information (vertices and faces) it contains. Let a box be just large enough to accommodate the model *bunny*, for example: the size of box is 100*100*100. The resolution of the box in Figure 7.3(a) is 10*10*10: hence, the box is divided into 1000 sub-boxes. The resolution of the box in Figure 7.3(b), by contrast, is 5*5*5: therefore, 8000 sub-boxes are included in the whole box. However, not every sub-box contains surface information, some of the sub-boxes may be empty. The higher the resolution, the more empty sub-boxes the bounding box will contain. To address this, we can simply remove the empty boxes. In our case, only the extracted salient regions need to be voxelized.

7.2.3 Points Selection

After the salient region has been voxelized, we may select one or more points from each voxel based on the entropy of the saliency values of vertices and restricted to the original model's surface. The entropy measures how the saliency of vertices in a voxel varies. The larger the variation, the larger the entropy, the more detail the voxel contains. Thus the entropy may be used to guide the points sampling.

For a voxel X, the entropy of each sub-box may be estimated by the finally estimated saliency values of each vertex:

$$H(X) = \sum_{i=1}^{n} - P(Saliency\_final(i)) * log_2(P(Saliency\_final(i))) \qquad (7.1)$$



(a) (b)

FIGURE 7.3: Example of voxelization with different resolutions on model *bunny*. (a) 1000 sub-boxes. (b) 8000 sub-boxes.

where $P(Saliency\_final(i))$ is the probability of a saliency value in a local voxel. In this work, the histogram is used to calculate the probability, and the number of bins are 30. n denotes the total number of the vertices in this voxel. Figure 7.4 shows the outline of the entropy estimation of vertices of each sub-box.

Finally, we choose one or more points from each sub-box, based on the estimated entropies. As aforementioned, the larger the entropy value, the more different saliency values are contained in that voxel. Therefore, we choose more vertices from the voxel with higher entropy, and fewer vertices from the voxel with lower entropy. The selection criterion is as follows: let N_{vox} be the number of voxels after voxelization, and let all the voxels be divided into two catergories: E_1 and E_2, where E_1 indicates the set of voxels with lower entropy values (voxel number: $\frac{N_{vox}}{2}$); E_2 indicates the set of voxels with higher entropy values (voxel number: $\frac{N_{vox}}{2}$). P_{req} denotes the total number of points are required to be sampled: then the number of points N_{E_1} and N_{E_2} are needed to be selected from each of voxel from the E_1 and E_2 sets, respectively, are:

$$N_{E_1} = \frac{0.5P_{req}}{N_{vox}} \tag{7.2}$$

$$N_{E_2} = \frac{1.5P_{req}}{N_{vox}} \tag{7.3}$$

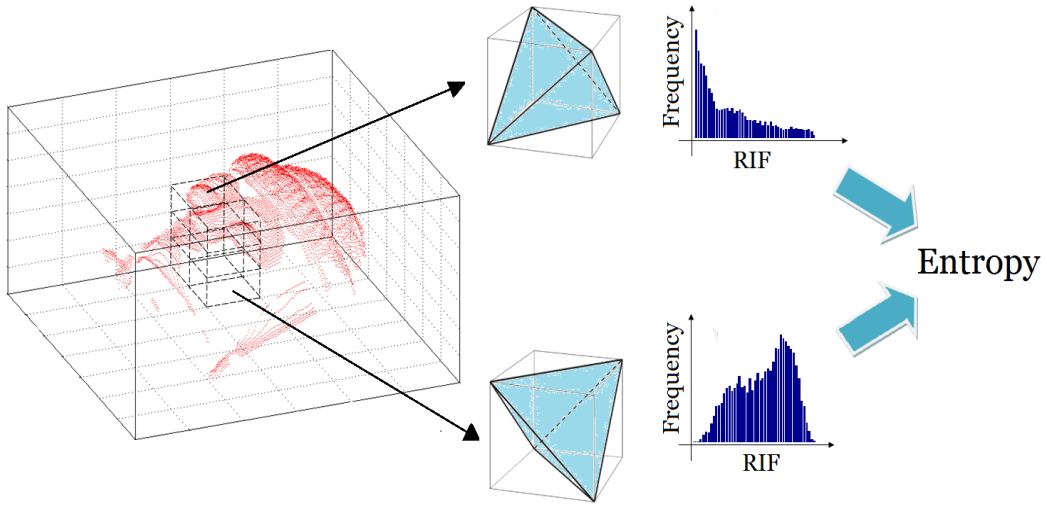


FIGURE 7.4: Outline of the procedure for computing the entropy of each voxel.

For example, if 200 vertices are required to be sampled, and 100 voxels exist after voxelization, we will sample 3 vertices each from each of the larger 50 voxels, and sample 1 vertex from the smaller 50 voxels. Normally, the vertex with maximum saliency value is chosen. Moreover, the minimum distances between the sampled vertices are used to address the possibility that the generated samples are close to the boundary between two or more adjacent voxels, or might be too close to each other. It is worth noting that the voxels were only divided into two categories in this work, and it is sufficient for the further application. However, future work will develop a metric to define the number of voxels categories, which could optimize the classification of the voxels.

7.3 Comparison Methods

In the following, we outline two state-of-the-art interest point detectors, which we reviewed in Chapter 2: 3D-Harris, and 3D-SIFT.

7.3.1 3D-Harris

Sipiran et al. [208] adopt the 2D corner detection method of Harris and Stephens. The extension method is based on first order derivatives along two orthogonal directions on the 3D surface.

1. A 2×2 matrix may be constructed for each vertex of the 3D mesh from the derivatives. The derivatives are obtained by fitting a quadratic surface to a neighborhood

of the vertex. The neighborhood around each vertex is defined and the centroid of the points in the neighborhood is calculated.

2. In order to achieve rotation invariance of the local coordinate frame, a PCA-based plane fitting is employed to the point set, and the eigenvector with the lowest eigenvalue corresponds to the normal of the fitting plane. The points set rotated, thus, the normal coincides with the z-axis of the local coordinate frame. As the vertex of analysis needs to coincide with the origin, the points are then re-translated.

3. After these transformations, a quadratic surface is fitted to the surface patch. The Harris operator values at all the vertices of the 3D mesh are calculated, and the local maxima are selected, taking into consideration the 1-ring neighborhood of a vertex. A vertex is determined as a local maxima if the Harris response of the given vertex is higher than those of all its immediate neighbors. However, only a constant fraction of the total number of these local maxima are selected as final interest points using the 1-ring neighbourhood.

7.3.2 3D-SIFT

The SIFT technique has also been adapted by Godil et al. [207] in to 3D. The authors described a 3D-SIFT operation on 3D voxel space. A scale space is constructed by applying 3D Gaussian filters with increasingly large scales to the voxelized model after voxelization, which is similar to the SIFT approach. A binary function is denoted on the voxelized model; then each layer of the scale space is represented by its convolution with a 3D Gaussian function. The Difference of Gaussian for each level is computed by subtracting the original model from the scaled model at the corresponding level. By searching the DoG space in both spatial and scale dimensions, the extrema points are detected. The extrema points are declared as interest points if they are located on the surface. Finally, the locations of the interest points are mapped back to the original mesh, and the closest vertices are marked as final interest points.

Note: In this experiment, for the purposes of fair comparison, we only consider the points that are locally distinctive to be points of interest in both the 3D-Harris and 3D-SIFT methods, as our VBS-guided and RBS-guided interest point detection only select points from salient regions. As a result, these points are able to show the local distinctive region, but lack the ability to represent the global shape. Thus, in this proposal, only the points with higher saliency will be designated as interest points: therefore, some portions of the surface do not have interest points. Furthermore, as we aim to claim that saliency-guided interest points detection is more accurate than the non-saliency guided interest

points detection, we only chose the interest points detection method (3D-Harris and 3D-SIFT), which are without saliency knowledge involved in this chapter for comparative study.

7.4 Evaluation Criteria: Repeatability

The most important quality of a keypoint detector is its repeatability. Repeatability rate is defined as the percentage of the detected points that are common in two different instances of a scene or objects. This characteristic accounts for the ability of the detector to find the same set of salient points on different instances of a given model, where the differences may be due to noise corruption, change of viewpoint, partial occlusions/missing parts, or a combination of these nuisances.

A repeatability criterion was first proposed by Schmid et al. [274], to be used to evaluate 2D keypoint detectors. Similarly, repeatability is also used in evaluation of 3D salient points detection. Repeatability is the geometrical stability of the corresponding interest points among multiple input data taken under varying conditions. The SHREC'10 [275] and SHREC'11 [276] robust feature detection and description benchmarks use the repeatability rate as the evaluation criterion under a variety of transformations. Once the correspondence of the points in a sequence has been established, we can define the repeatability rep in the frame i of a sequence as:

$$rep = \frac{|np \cap np_r|}{|np_r|} \tag{7.4}$$

where np is the number of interest points found from a model captured from another viewpoint, or noise-corrupted by different methods, and np_r is the number of interest points detected from a model captured from a particular viewpoint, or original model (reference model). For a perfect detector, $rep = 1$: i.e., it detects the same interest points in the first and last viewpoints.

In order to evaluate the repeatability of our detected interest points, we investigate our saliency-guided interest points detection approach under different variations of input data: change of viewpoint, and noise corruption. The same as the saliency-guided simplification, the input data are also from the OSU and the SRID. Our saliency-guided interest points detection method is also compared with two state-of-the-art 3D surface point detectors: 3D-Harris [208], and 3D-SIFT [207]. In addition, in order to achieve a fair comparison, we select the 1% number of vertices as interest points from the original object. In practice, we have already collected the coordinates of each vertex from whole

3D model. Thus, whether the two points are repeatable may be determined from the coordinates information of detected points under two instances of data transformation.

Figure 7.5 shows the outcomes of the different 3D interest point detectors on models *buddha*, *rick-face*, *duck* and *frame* as detected by four different methods. From the standpoint of visual appearance, each of the methods has the capacity to detect points from the areas with a high density of local details. As we have no ground truth of interest point detection with which to measure their relative accuracy of detection, we cannot determine whether our method achieves results comparable to those of its competitors. Instead, in order to evaluate the proposed and other interest point detectors, we will study the degree of repeatability of their detection of interest points under the conditions of change of viewpoint and noise corruption.

FIGURE 7.5: Interest points detection employing different methods. (a) 3D-SIFT; (b) 3D-Harris; (c) VBS; (d) RBS.

7.4.1 Effect of Change of Viewpoint

This experiment evaluates the susceptibility of the proposed detector and its competitors in response to change of viewpoint. Models *buddha*, *lobster* and *duck* were subjected to a rotation of θ at intervals of 20° around an unknown rotation axis, and model *pat-face* was subjected to a 36° rotation angle. We set one of the images as the reference image, and calculated the repeatability of interest points from four alternative viewpoints, of which two viewpoints were anti-clockwise rotated and two viewpoints were clockwise rotated. Figure 7.6 and Figure 7.7 show the results of these rotations of model *lobster* and *dinosaur* from different range image datasets.

In this section, the following datasets were selected to validate the robustness of the proposed interest points detection method and its competitors (reference viewpoints are in underline and bold type):

A. OSU range image dataset:

- *buddha*: 320°, 340°, **0°**, 20°, 40°.
- *lobster*: 60°, 80°, **100°**, 120°, 140°.
- *duck*: 320°, 340°, **0°**, 20°, 40°.

B. SRID range image dataset:

- *dinosaur*: view 31, view 32, **view 33**, view 34, view 35.
- *isis*: view 34, view 35, **view 36**, view 37, view 38.
- *copter*: view 16, view 17, **view 18**, view 19, view 20.

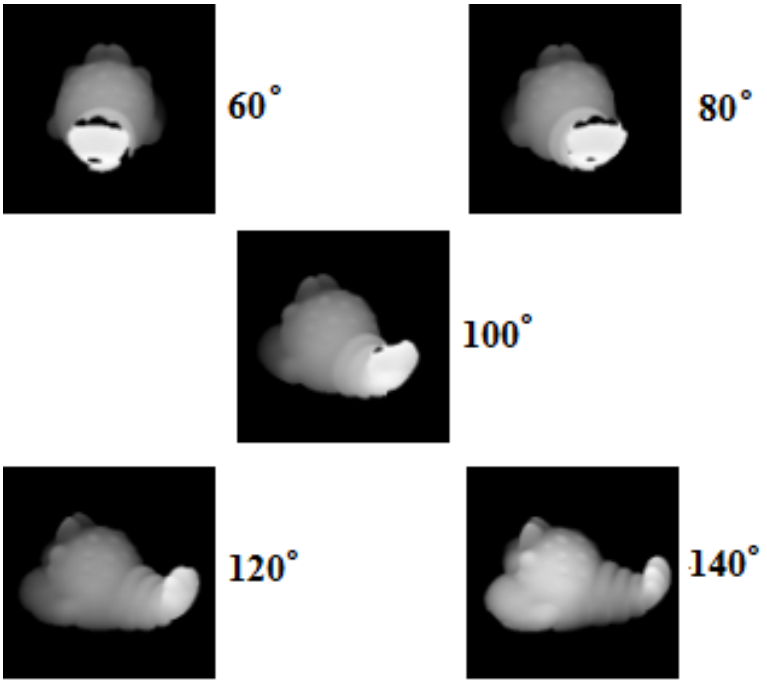


FIGURE 7.6: Sample of five scans of model *lobster* from OSU database, subjected to rotations at intervals of 20°, from 60° to 140° around an unknown rotation axis followed by a translation. The middle scan is set as the reference scan (100°). Two scans were anti-clockwise rotated and two scans were clockwise rotated.

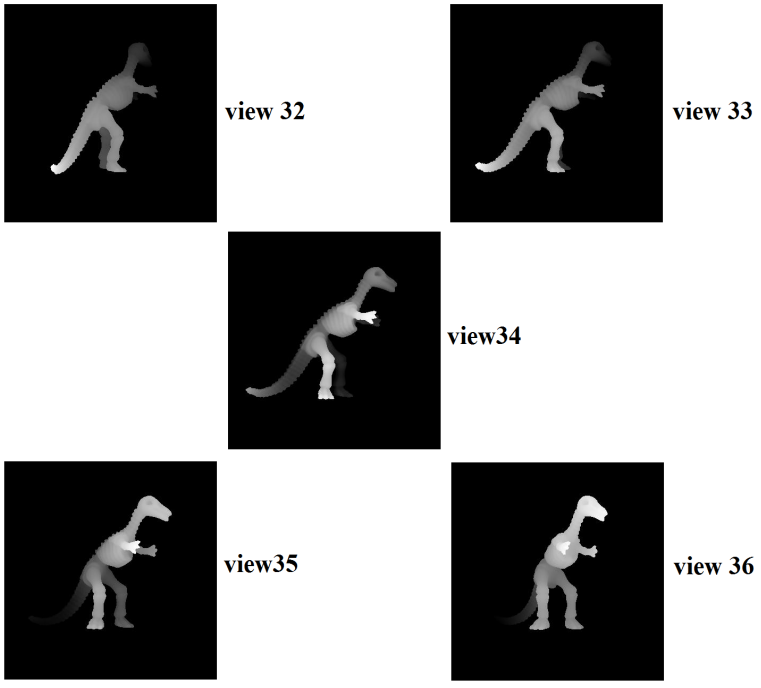


FIGURE 7.7: Sample scans of object *dinosaur* from the SRID database with different viewpoints. The middle scan is set as the reference scan (view 34). Two scans were anti-clockwise rotated, and two scans were clockwise rotated around an unknown rotation axis followed by a translation.

Figure 7.8 - Figure 7.10 and Figure 7.12 - Figure 7.14 show the comparison of our methods with 3D-Harris and 3D-SIFT under change of viewpoint in two different datasets. Each of the methods has the capability to detect points from the areas with a high density of local details. The top two rows are the detected interest points as found using VBS and RBS saliency detection. It may be seen that the interest points remain consistently detected, such as the mouth and eyes region of *buddha* in Figure 7.8, bumpy areas of *lobster* in Figure 7.9, wing region of *duck* in Figure 7.10, extremities of *dinosaur* Figure 7.12, face of *isis* Figure 7.13 and propeller blade of *copter* in Figure 7.14 because the salient regions are steadily detected under different rotation angles by both the VBS and RBS saliency detection methods. However, on careful observation of model *buddha* in Figure 7.8, it will be noted that, in the case of the VBS approach, only a few points are detected as of interest in the mouth region, as this region is detected as a salient discontinuity: according to our points selection criteria, this method may fail to detect the interest points when the resolution of voxelization is low.

3D-Harris and 3D-SIFT may not detect repeatable points under change of viewpoint due to different points detection criteria: this is shown in the bottom two rows of Figure 7.8 - Figure 7.10 and Figure 7.12 - Figure 7.14. In order to illustrate the ability of these four methods to detect the same point under different transformations, the repeatabilities are demonstrated.

Figure 7.11 and Figure 7.15 show the repeatabilities in response to viewpoint transformation. Most detectors show excellent tolerance with small degrees of rotation of viewpoint, such as 20°. For the model *buddha*, RBS achieves repeatability rates around 0.8, as shown in Figure 7.11: VBS produces a 0.75 repeatability rate, 3D-SIFT a 0.6 repeatability rate, and 3D-Harris only 0.43.

Looking at the repeatabilities of model *dinosaur* in Figure 7.15, it may be seen that, overall, on the curves of the figure, the RBS produces the most stable results, the difference of repeatability between reference and neighboring views being only around 0.2. By contrast, in the case of the 3D-Harris and 3D-SIFT detectors, the curve changes dramatically with change of detector, the difference of repeatability between these two neighboring views being around 0.35.

Moreover, with increasing rotation angles of viewpoint, repeatability rates fall, since the larger the rotation angles of viewpoint, the smaller the number of overlapped points. For instance, in Figure 7.11 we see that RBS is able to find approximately 70% of the overlapping points in the model with a rotation angle of viewpoint difference of 40° on model *lobster*. The second-best performer is VBS, which achieves an accuracy of 60%, as much as 10% lower than RBS. The repeatability rate of 3D-Harris is 0.55, which is 15% lower than RBS and 5% weaker than VBS. This is because 3D-Harris relies heavily

on the selection of parameters,, such as the number of rings around each vertex, and the constant fraction of the total number of vertices with the highest Harris response. It proved to be difficult to find the appropriate parameters during our reimplementation: as a result, the rate at which interest points were falsely detected was increased.

The worst-performing approach in this case is 3D-SIFT, which reports only 55% of overlapping points - a reduction of almost 30% compared with RBS. Because it makes use of a coarse voxel structure, 3D-SIFT sometimes locates interest points onto insignificant regions. Moreover, the strategy of 3D-SIFT seeks to avoid detecting points from the boundaries and edges: as a result, it did not well-localize the finer interest points. In summary, the proposed method achieves the best performance under change-of-viewpoint transformation.

(a) (b) (c)

FIGURE 7.8: Model *buddha* subject to a rotation of θ, at intervals of 20° and 40° around an unknown rotation axis followed by a translation. These transformations are used to test the robustness of the proposed method when applied to interest points selection. Top row: RBS. Second row: VBS. Third row: 3D-Harris. Bottom row: 3D-SIFT. (a) Reference viewpoint; (b) 20° rotated; (c) 40° rotated.

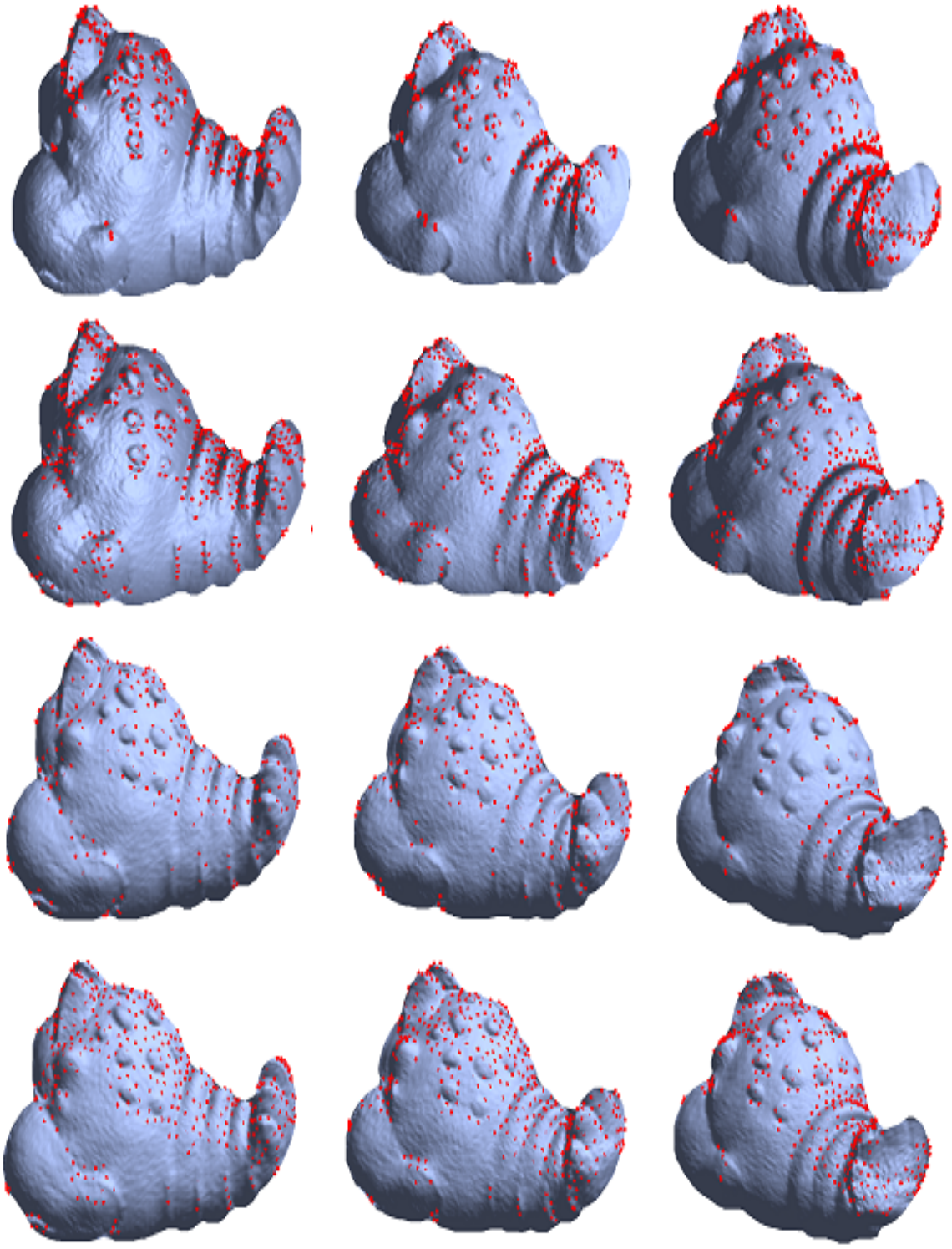
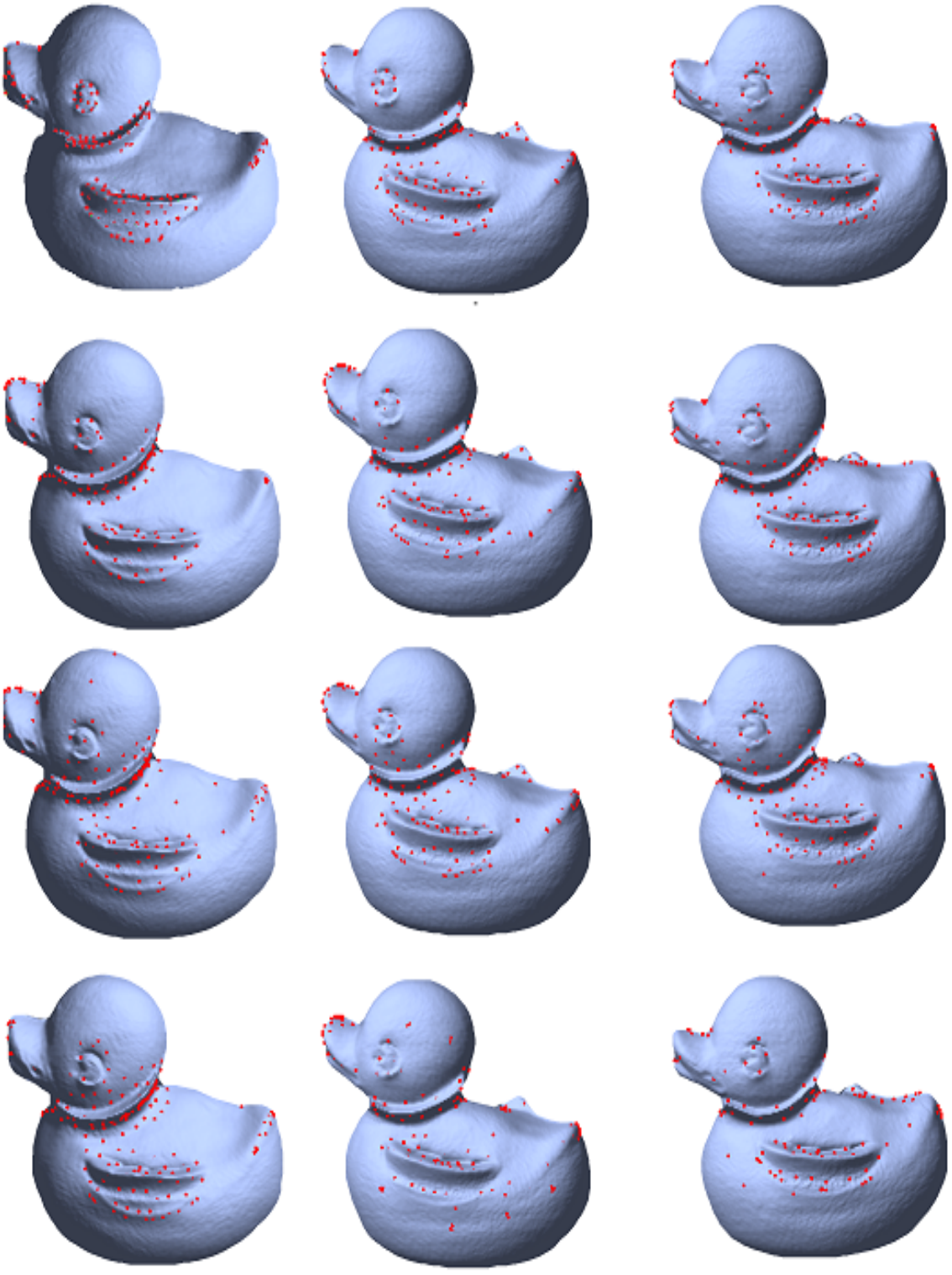


FIGURE 7.9: Model *lobster* subject to a rotation of θ, at intervals of 20° and 40° around an unknown rotation axis followed by a translation. These transformations are used to test the robustness of the proposed method when applied to interest points selection. Top row: RBS. Second row: VBS. Third row: 3D-Harris. Bottom row: 3D-SIFT. (a) Reference viewpoint; (b) 20° rotated; (c) 40° rotated.



(a) (b) (c)

FIGURE 7.10: Model *duck* subject to a rotation of θ, at intervals of 20° and 40° around an unknown rotation axis followed by a translation, are used to test the robustness of the proposed method when applied to interest points selection. Top row: RBS. Second row: VBS. Third row: 3D-Harris. Bottom row: 3D-SIFT. (a) Reference viewpoint; (b) 20° rotated; (c) 40° rotated.

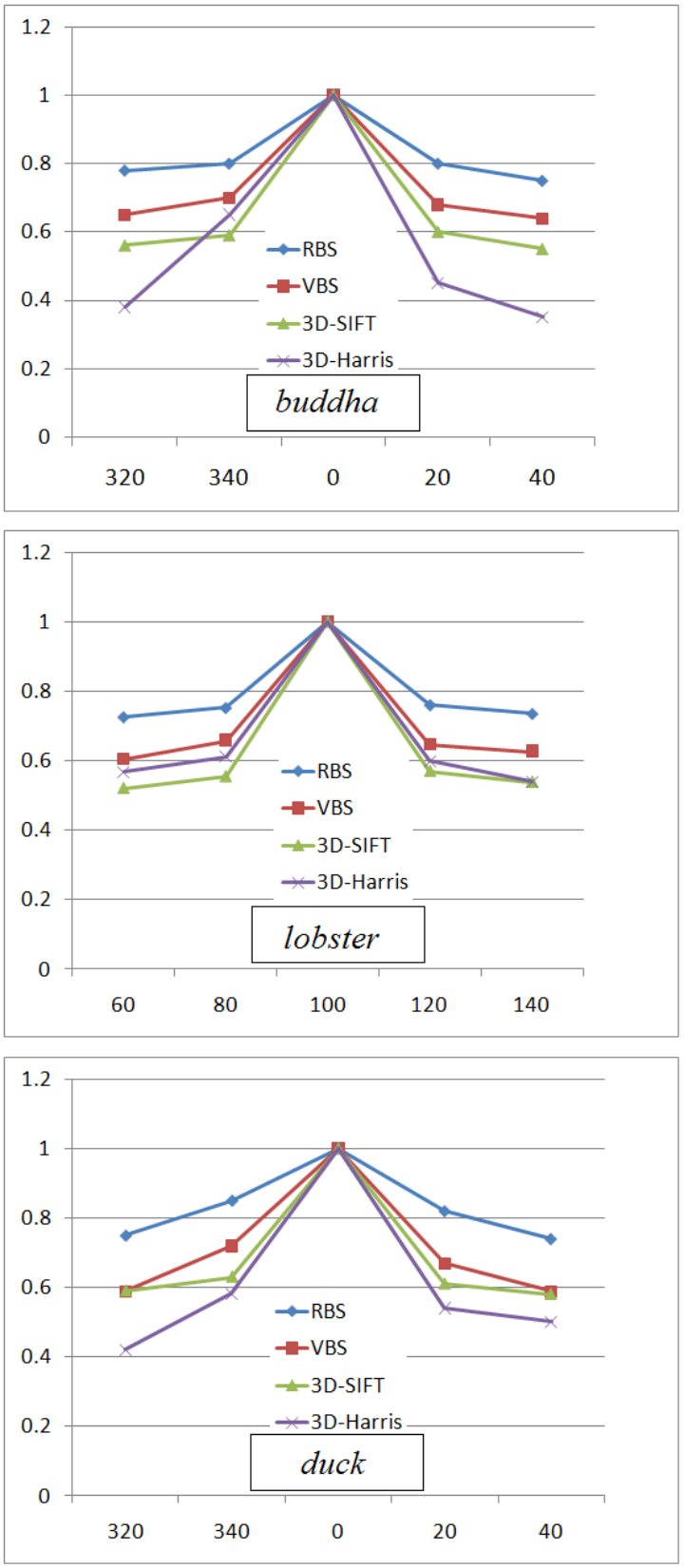


FIGURE 7.11: Repeatability of interest points on OSU models with different detectors under different rotation angles of θ: $\pm 20°$ and $\pm 40°$. From top to bottom: *buddha*; *lobster*; *duck*.

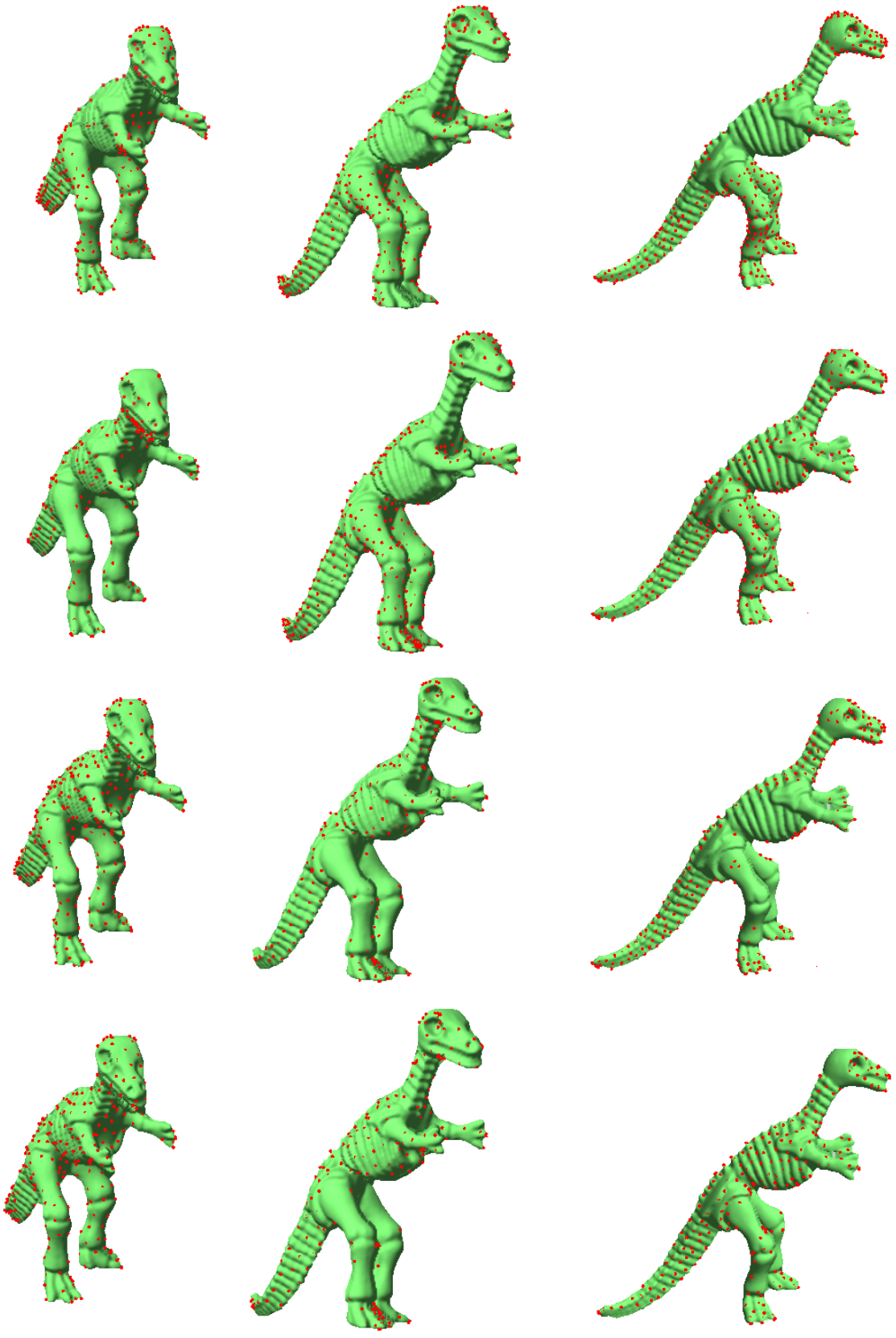


FIGURE 7.12: Model *dinosaur* imaged from three different viewpoints: (left to right in each row) view 31, view 32, and view 33. View 33 is set as the reference view in our experiment. The other views were captured by a rotation of the reference view with of 20° and 40° around an unknown rotation axis followed by a translation. Top row: RBS. Second row: VBS. Third row: 3D-Harris. Bottom row: 3D-SIFT.

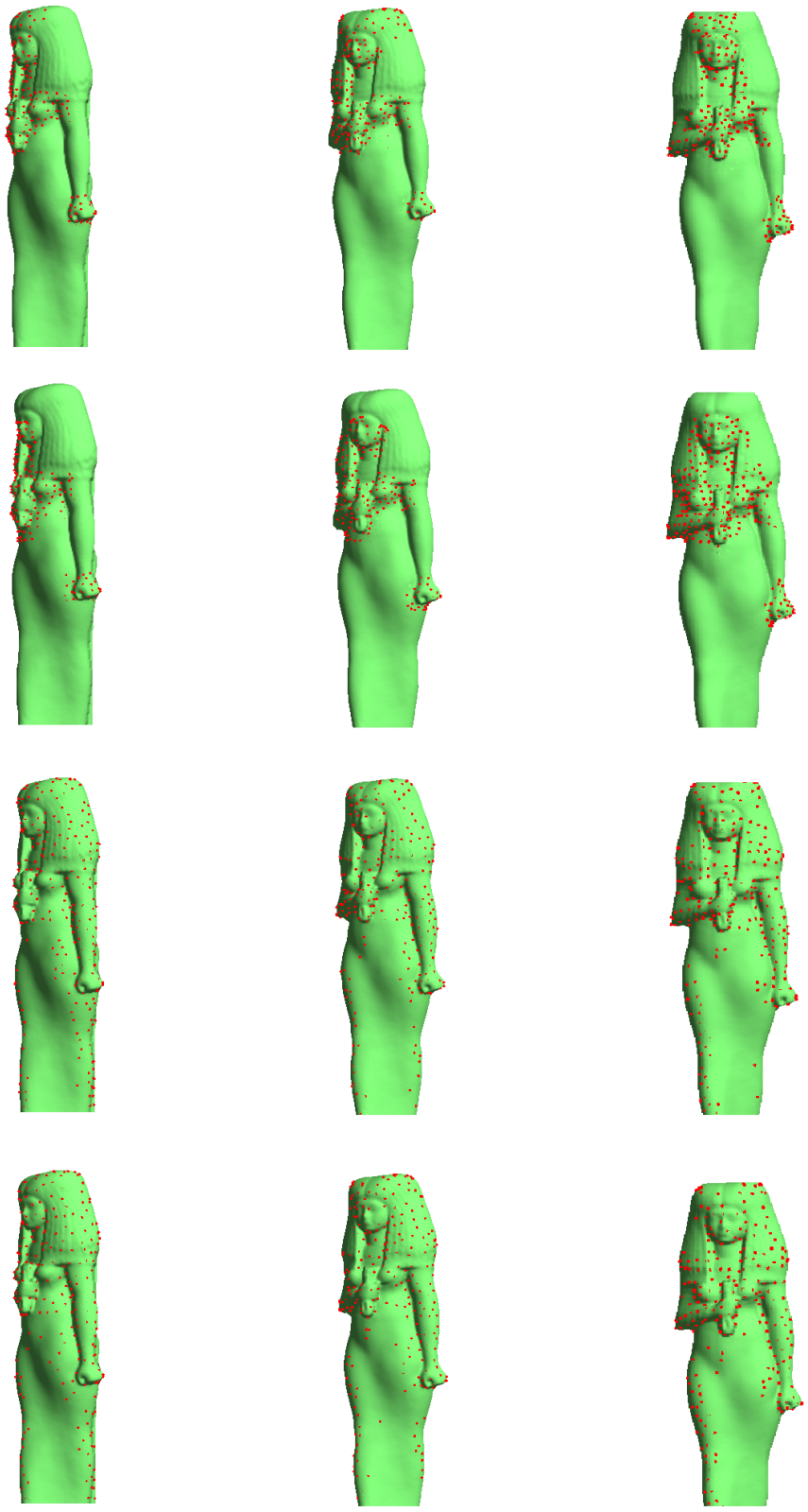


FIGURE 7.13: Model *isis* imaged from three different viewpoints: view 34, view 35, and view 36. View 36 is set as the reference view in our experiment. The other views were captured by a rotation of the reference view of 20° and 40° around an unknown rotation axis followed by a translation. Top row: RBS. Second row: VBS. Third row: 3D-Harris. Bottom row: 3D-SIFT.

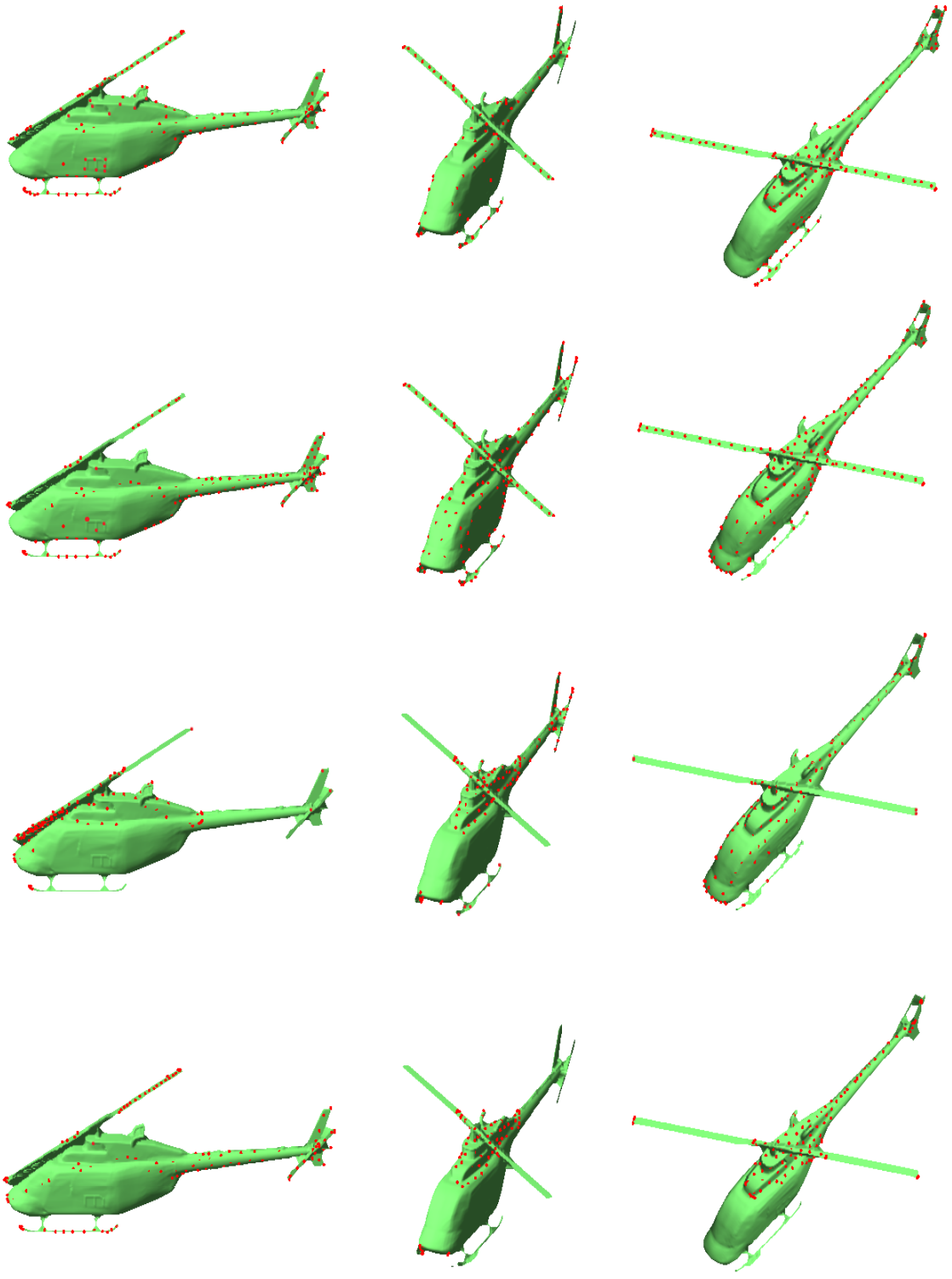


FIGURE 7.14: Model *copter* imaged from three different viewpoints: view 16, view 17, and view 18. View 18 is set as the reference view in our experiment. The other views were captured by a rotation of the reference view of 20° and 40° around an unknown rotation axis followed by a translation. Top row: RBS. Second row: VBS. Third row: 3D-Harris. Bottom row: 3D-SIFT.

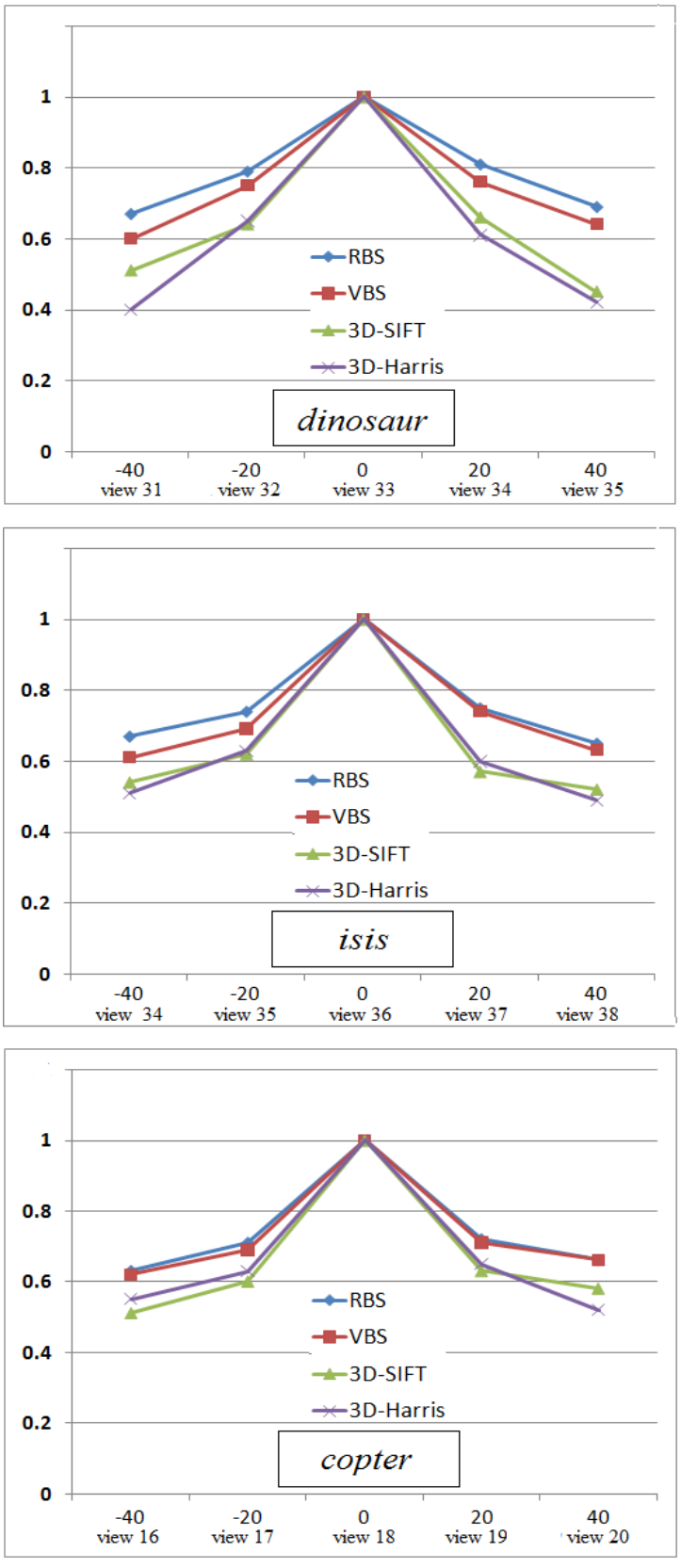


FIGURE 7.15: Repeatability of interest points on SRID models with different detectors under different rotation of θ: $\pm 20°$ and $\pm 40°$. From top to bottom: *dinosaur*; *isis*; *copter*.

7.4.2 Effect of Noise Corruption

In order to validate the robustness of our saliency-guided interest points detection methods, different levels of random Gaussian white noise with a standard deviation of σ were added to the original data. After the surface is noise-corrupted, the local topology around a vertex changes considerably. As a result, the regions of a noisy surface detected as salient may differ from those detected from the uncorrupted original. However, it may be seen that our method can still distinguish salient from non-salient regions in the presence of a considerable amount of noise. After comparison, it may be observed that the regions estimated as salient from noisy data and clean data remain almost the same.

Figure 7.16 - Figure 7.18 and Figure 7.20 - Figure 7.22 illustrate the interest points detected by the RBS- and VBS-guided methods and other two comparitive methods 3D-Harris and 3D-SIFT, from different levels of noise corruption of the surface. Figure 7.16 - Figure 7.18 indicate interest points detection on the OSU dataset. All of the methods are able to detect the interest points from the local distinctive regions with low level of noise: $\sigma = 0.1$. However, the 3D-Harris and 3D-SIFT start to detect points from non-salient region as interest when the noise level is increased to $\sigma = 0.2$ and $\sigma = 0.3$. For example, in Figure 7.16, the 3D-Harris select point from the forehead are of the *buddha* with $\sigma = 0.3$, and in the *lobster* case of Figure 7.17, the bumpy area are also missed points selection. In Figure 7.18, the 3D-Harris and 3D-SIFT detectors select the points as interest from the flatter region of *duck*, but lack of selection from the portion where should be more important, the eye. Figure 7.20 - Figure 7.22 illustrate the results on the SRID dataset. All of the detectors show excellent tolerance with low level noise corruption. With increasing the noise level, the detected interest points by 3D-Harris and 3D-SIFT are starting to distribute on the non-salient region, such as the skirt area of *isis* in Figure 7.21. However, it can be senn that our methods, especially RBS, still can select interest points from the local distinctive area, such as the eye region of *dinosaur* in Figure 7.20, propeller blade of *copter* in Figure 7.22.

In order to show the robustness of our saliency-guided points detectors under different levels of noise corruption, the repeatability rates of their interest points are also reported, in Figure 7.19 and Figure 7.23. These results were obtained by adding different levels of Gaussian noise: as the Gaussian noise is generated randomly, we ran the experiment 10 times at each noise level, and report the average repeatability rate. As we expected, the presence of noise affects repeatability: as the σ increases, the repeatability rates decrease for all the detectors. For example, the case of model *buddha* in Figure 7.19 shows that when $\sigma = 0.1$, the surface is only slightly corrupted by noise, and the 3D-SIFT achieves

a high repeatability rate of 0.7: but this is still inferior to that of RBS and VBS, whose rates are 0.73 and 0.75. 3D-Harris produces the lowest rate in this case, at only 0.6.

Figure 7.23 demonstrates the repeatability rate of *copter*. The rates achieved by VBS, RBS, and 3D-SIFT are very similar at all three of these noise levels. 3D-Harris performs worst, giving results 25% lower than RBS when $\sigma = 0.1$, 21.4% lower than RBS when $\sigma = 0.2$, and 36.5% lower than RBS when $\sigma = 0.3$.

Overall, 3D-SIFT presents a relatively weaker tolerance than the proposed methods, since 3D-SIFT works on a coarse voxel structure, and sometimes locates interest points from insignificant regions. Moreover, a scalar function is built in 3D-SIFT: this scalar function is obtained by mean and Gaussian curvature, and the curvature is known to be sensitive to noise. 3D-Harris aims to find areas of high curvature, and tends in consequence to find fewer features and more sharp corners [208]. In addition, 3D-Harris does not locate interest points over flat or smooth regions, but is excessively sensitive to corners and ridges, producing multiple interest points around the corners of the object [208]. In general, corner detection approaches are relatively less robust to noise.

(a) (b) (c)

FIGURE 7.16: Interest points detected at different noise levels on model *buddha*. From left to right: (a) $\sigma = 0.1$; (b) $\sigma = 0.2$; (c) $\sigma = 0.3$. From top to bottom: RBS-based method; VBS-based method; 3D-Harris; and 3D-SIFT.

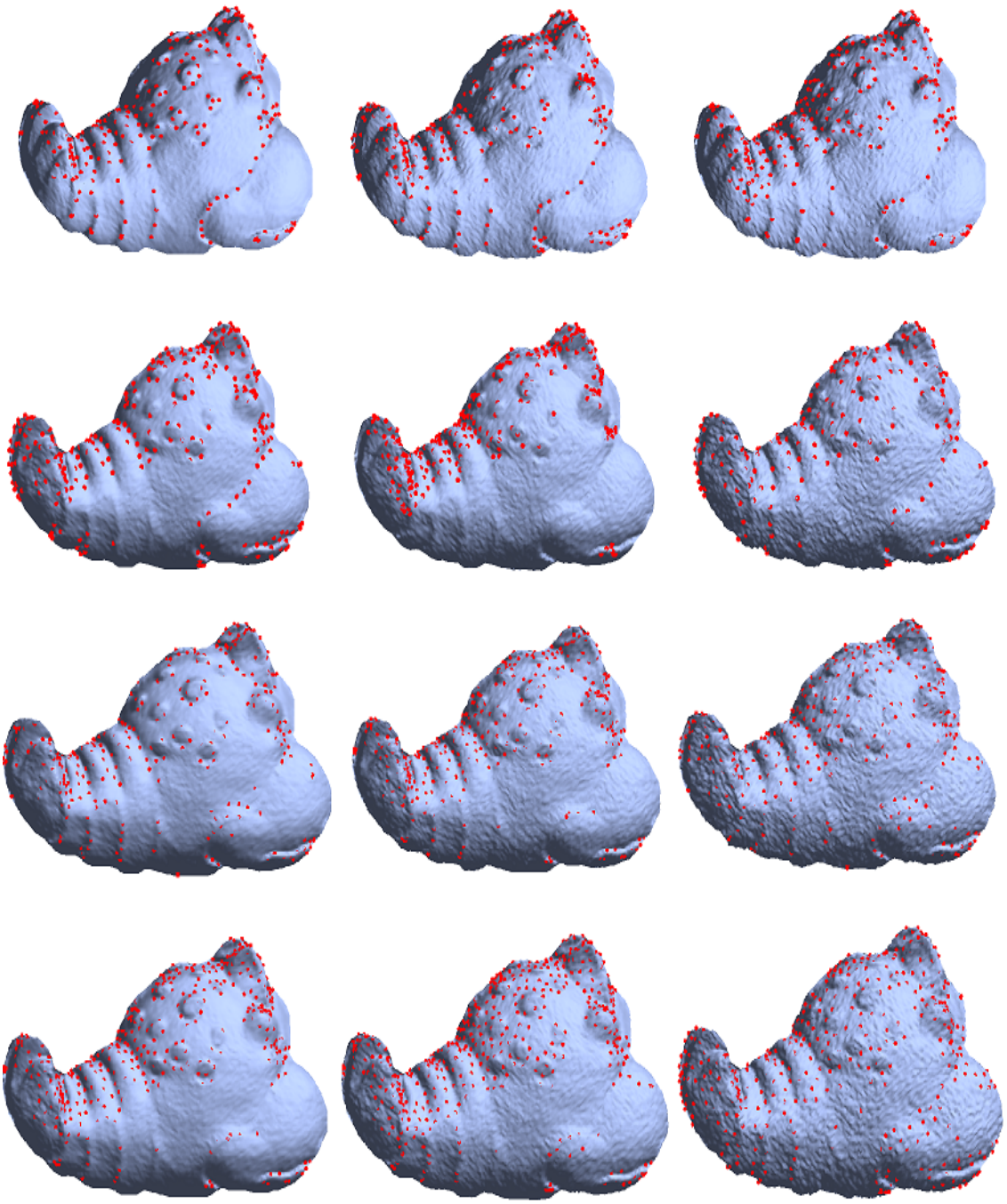


FIGURE 7.17: Interest points detected at different noise levels on model *lobster*. From left to right: (a) $\sigma = 0.1$; (b) $\sigma = 0.2$; (c) $\sigma = 0.3$. From top to bottom: RBS-based method; VBS-based method; 3D-Harris; and 3D-SIFT.

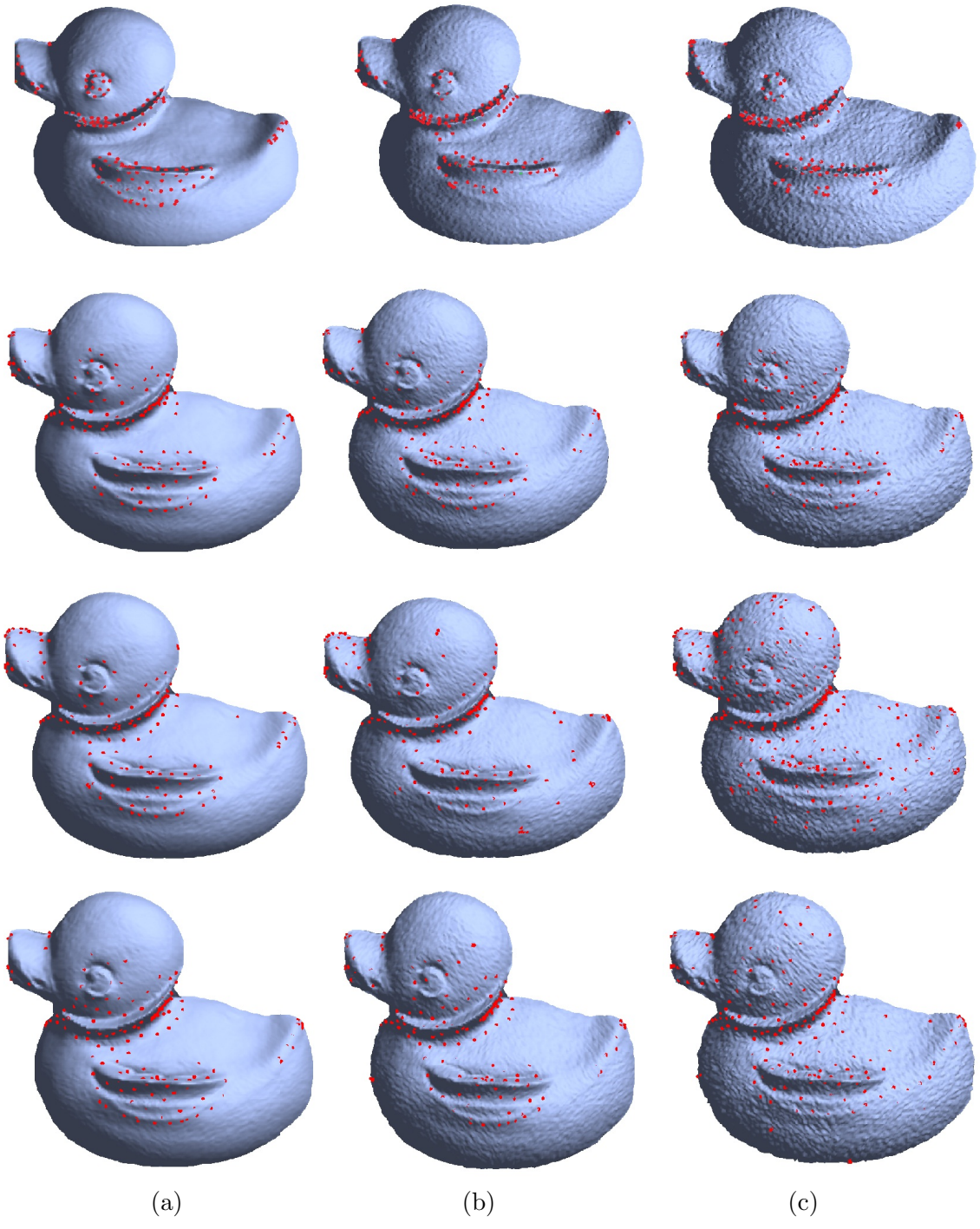


FIGURE 7.18: Interest points detected at different noise levels on model *duck*. From left to right: (a) $\sigma = 0.1$; (b) $\sigma = 0.2$; (c) $\sigma = 0.3$. From top to bottom: RBS-based method; VBS-based method; 3D-Harris; and 3D-SIFT.

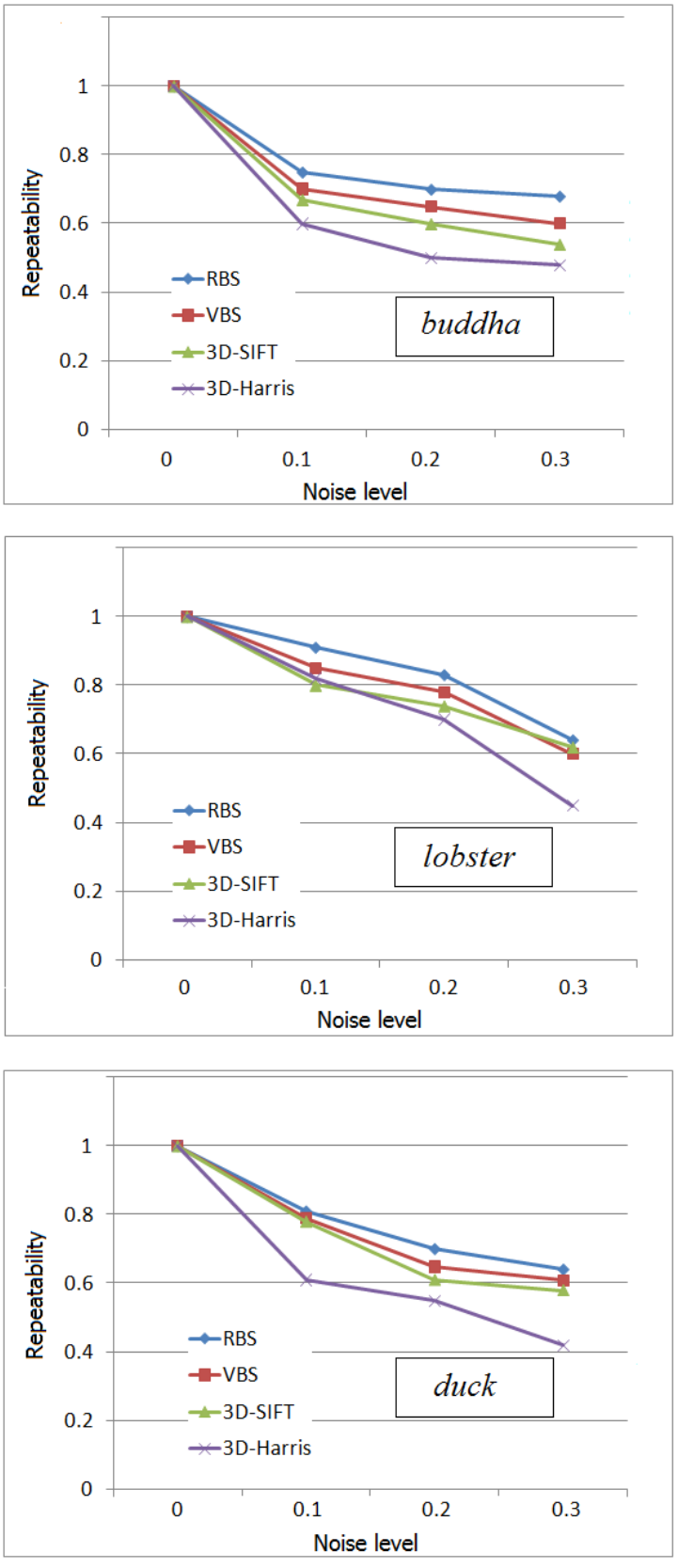


FIGURE 7.19: Repeatability of interest points on OSU models, respectively, with different detectors applied under different noise levels. From top to bottom: *buddha*; *lobster*; *duck*.

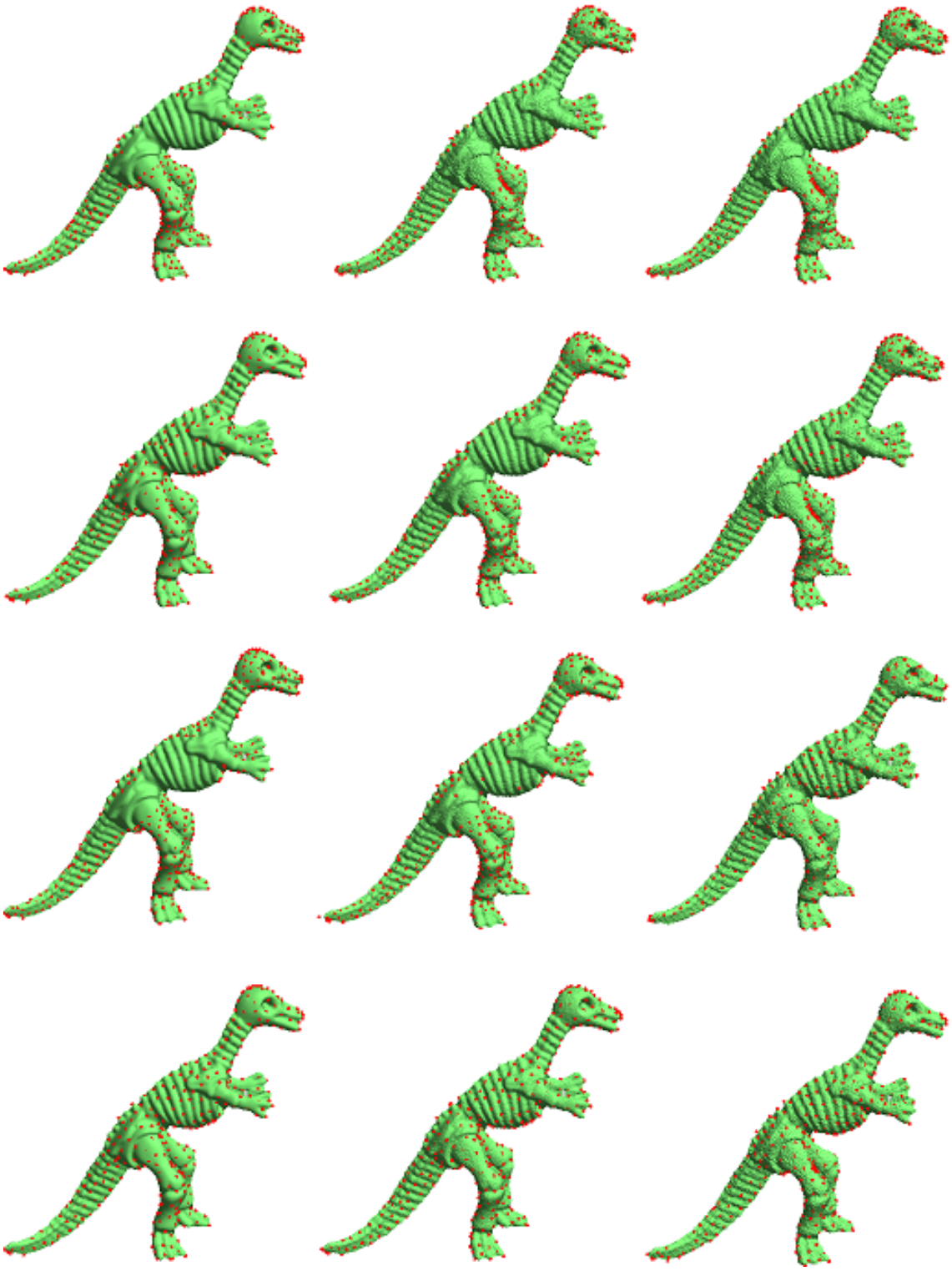


FIGURE 7.20: Interest points detected at different noise levels on model *dinosaur*. From left to right: (a) $\sigma = 0.1$; (b) $\sigma = 0.2$; (c) $\sigma = 0.3$. From top to bottom: RBS-based method; VBS-based method; 3D-Harris; and 3D-SIFT.

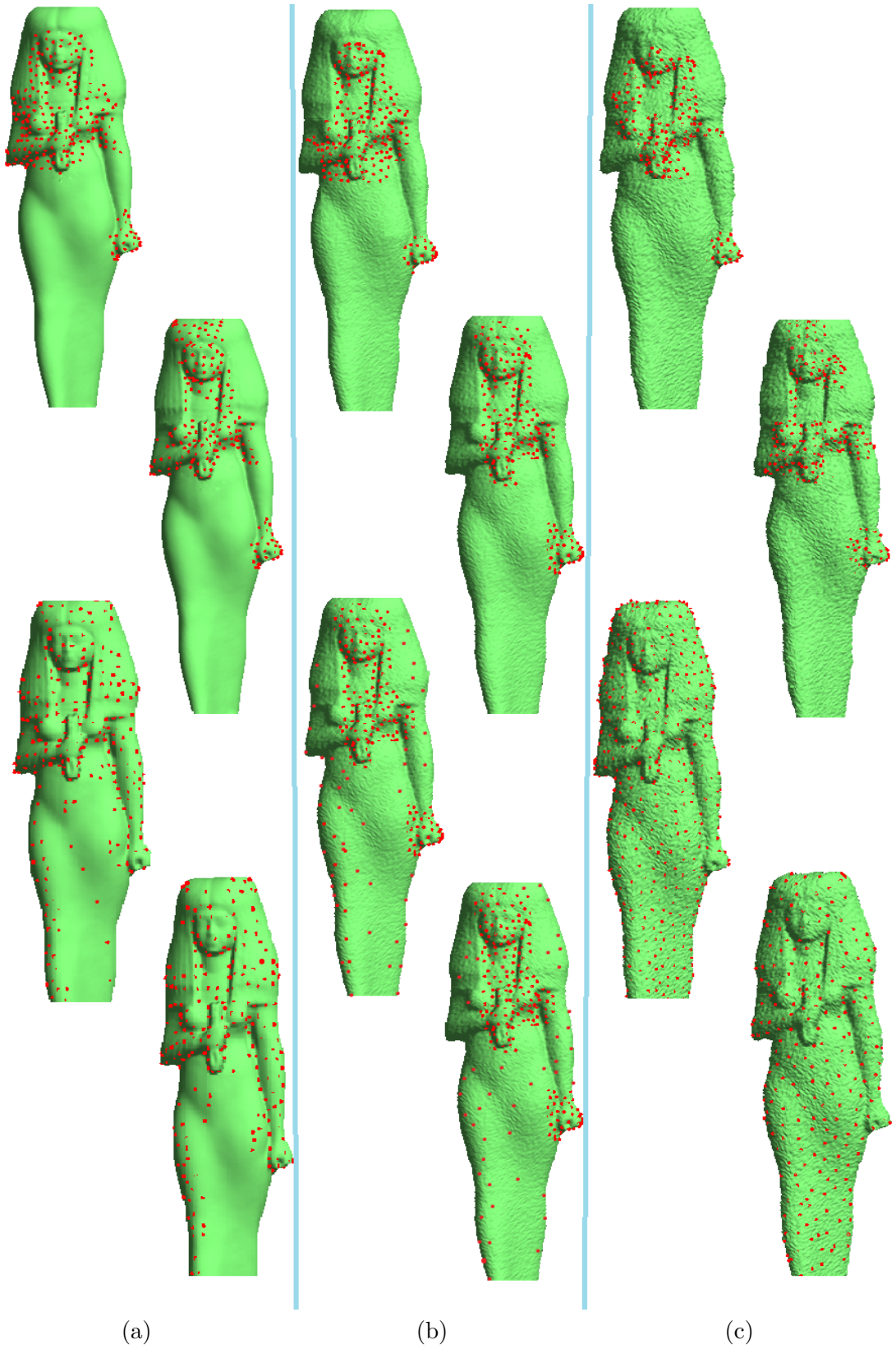
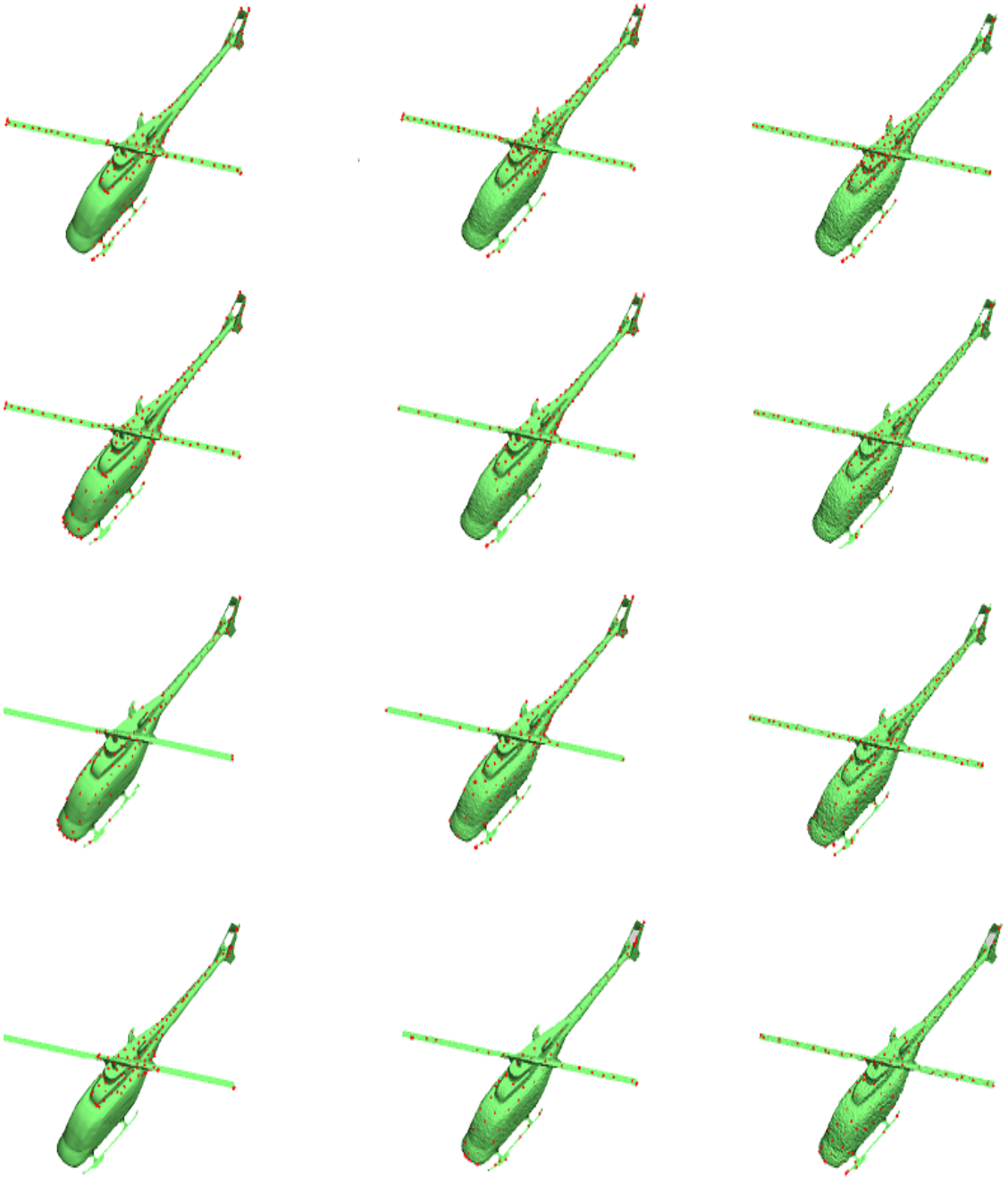


FIGURE 7.21: Interest points detected at different noise levels on model *isis*. From left to right: (a) $\sigma = 0.1$; (b) $\sigma = 0.2$; (c) $\sigma = 0.3$. From top to bottom: RBS-based method; VBS-based method; 3D-Harris; and 3D-SIFT.



(a) (b) (c)

FIGURE 7.22: Interest points detected at different noise levels on model *copter*. From left to right: (a) $\sigma = 0.1$; (b) $\sigma = 0.2$; (c) $\sigma = 0.3$. From top to bottom: RBS-based method; VBS-based method; 3D-Harris; and 3D-SIFT.

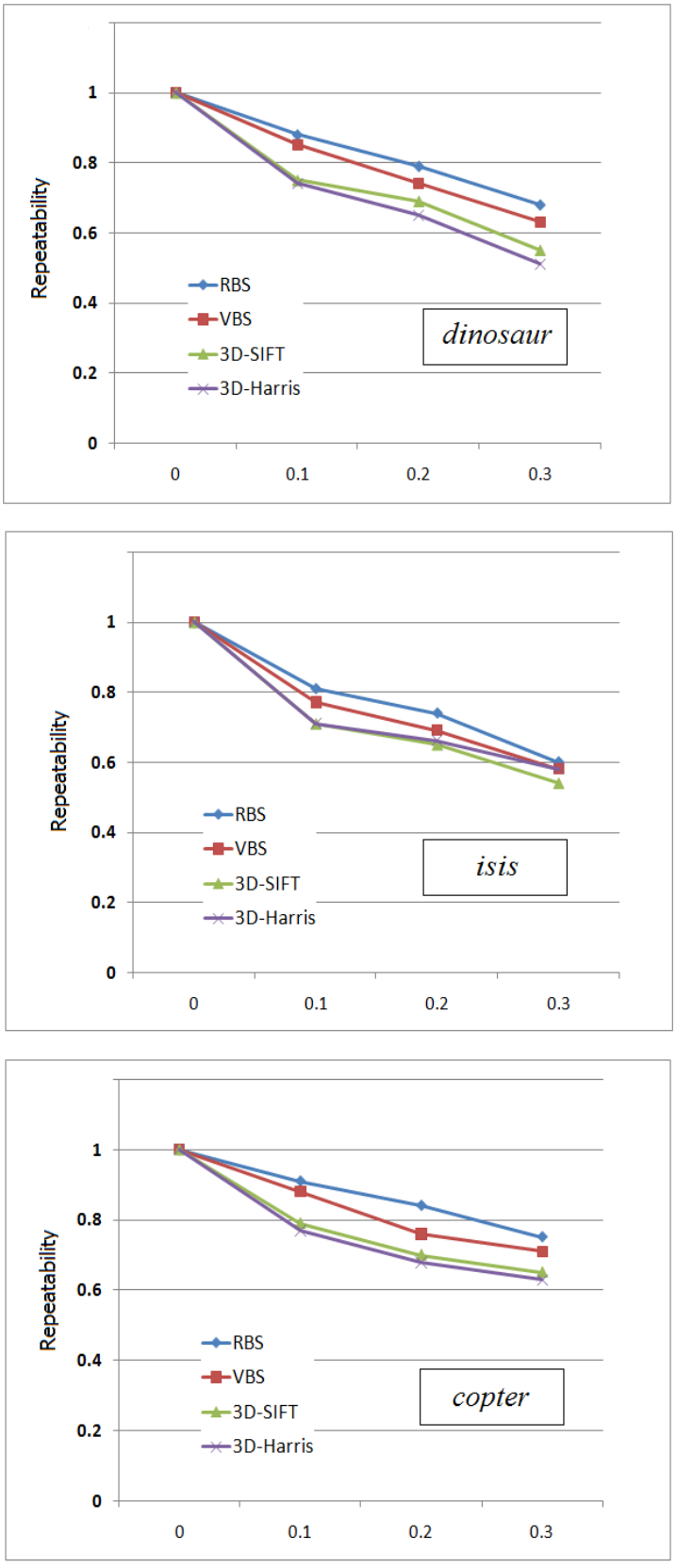


FIGURE 7.23: Repeatability of interest points on SRID models, respectively, with different detectors applied under different noise levels. From top to bottom: *dinosaur*; *isis*; *copter*.

7.5 Summary

In the case of two saliency-guided applications, we have analysed their interest point detection performance. We have proposed a simple but effective points detector, which operates in three steps: extraction of the salient regions; voxelization of the extracted salient regions; and point selection determined by the entropy of each voxel.

The 3D-Harris and 3D-SIFT methods were re-implemented for purposes of comparison with the proposed saliency-guided points detectors. The repeatability rate was employed to illustrate the ability of the detectors to find the same set of points on different instances of a given model. In our project, the differences were due to noise corruption and change of viewpoint. Comparison of the repeatability rates clearly showed that the performances of 3D-Harris and 3D-SIFT are affected by rotation and noise. The same interest points may hardly be detected again after such data transformations. It is noticeable that both evaluation methods agree in indicating that the proposed saliency-guided points detectors are the most suitable detectors in the context of the experiments that have been performed. The 3D-Harris is not invariant to rotation - which is a limitation in most 3D applications. 3D-SIFT is sensitive to noise corruption. Overall, the RBS saliency-guided detector has the best performance in the case of both noise corruption and change of viewpoint. That is because the proposed saliency detection is robust to both forms of data transformation (see Chapter 6), and the points detector relies purely on the salient regions.

Chapter 8

Conclusion

In this final chapter, we present the summary of this thesis and the conclusions drawn from the presented work. The main contributions and novel aspects of our research are summarised. At the end of this chapter, a list of our publications related to this thesis is provided.

8.1 Summary

In this dissertation, we explored techniques for estimating the saliency region from different 3D database models. The whole work was divided into four phases: surface smoothing, Retinex-based importance feature estimation, saliency detection, and saliency-guided applications. Our main focus has been on detecting the important, salient regions of 3D models, and proposing 3D applications based on those detected salient regions.

We started with a brief introduction of the research topics under investigation in this thesis: the background of the 3D imaging and data representation techniques, the challenges we are facing in the form of the massive increases in data size, and the way to address these problems - a *saliency-guided* method. A review of the relevant literature was carried out in the following chapter. As the proposed project is a framework which combines multiple 3D techniques, we studied the existing approaches of 2D/3D saliency detection, surface smoothing, segmentation, simplification, and interest points detection. From these reviews, we summarised the research challenges and concluded that it is still well worthwhile to develop more effective and robust methods for each step.

In order to build a complete framework for saliency detection on 3D surfaces, we first developed an extended non-local means filter-based smoothing method as a pre-processing

step (Chapter 4). This smoothing algorithm has the ability to remove noise and smooth the surface, but retains the local features. Subsequently, in order to develop a more effective feature for saliency detection, we proposed the Retinex-based Importance Feature (Chapter 3) and Relative Distance metric (Chapter 5), which were obtained from the geometric and perceptual properties of the model. For saliency detection on 3D shapes, we developed two methods: vertex-based saliency detection (VBS) and region-based saliency detection (RBS). The VBS algorithm was inspired by the *mesh saliency* method. However, two new feature channels were estimated: RIF and RD. This method not only takes into consideration the geometry of the surface, but also takes into account the vagaries of human visual perception. The RBS algorithm is an extension beyond the VBS, as the detection of vertex-based saliency is affected by occlusion and holes. The RBS algorithm consisted of local and global information to estimate the saliency.

Finally, the detected salient regions were used as important factors for two 3D applications: simplification, and interest points detection. In the simplification application, the resulting simplified surfaces, which were guided by salient regions detected by the two proposed methods, had well-preserved the important local features. From the viewpoint of mesh errors, smaller RMSE and Metro errors were achieved by the proposed methods. In the case of the application for the detection of interest points, the proposed methods have the capability to select the interest points robustly under different interruptions from external factors. The repeatability rates also showed that the proposed methods are more stable and effective than the competitors.

8.2 Contributions

The main contributions and novel aspects of this thesis are summarised as follows:

First, we have presented an extensive literature review on 3D techniques that were involved in our saliency detection work. This includes 2D and 3D saliency detection, smoothing, Retinex-based image enhancement, simplification, and interest points detection. The main points of all the reviewed approaches were described and summarised.

Second, a new smoothing method for 3D surfaces was developed by extending the non-local means filter, which was originally applied only to 2D images. 2D images usually have a regular structure, which in most cases is not true for a mesh, due to variations of sampling density in the range scanning process. Therefore, in our work, we employ an alternative approach to define a regular structure in order to obtain the similarity kernel of non-local means on a 3D mesh: the B-Spline surface. The advantage of using B-spline surfaces is that the underlying control net is topologically similar to the image

grid structure. The proposed smoothing methods well-preserve the local details and global shapes. The mesh errors between original and smoothed versions were also much lower than was the case for the competitive methods. The corresponding work has been published in [225].

Third, two new feature channels for saliency estimation were proposed: relative distance (RD), and Retinex-based importance feature (RIF). Most saliency detection systems require one or more features from the 2D image or 3D shape to generate the saliency, such as the color, intensity, orientation, curvature, or shape index. The RD and RIF are based on the surface geometry, and human visual perception respectively. As a measure, RD has the ability to describe the local type of shape as a continuous parameter, and has the ability to represent the concave and convex rejoin significantly. RIF combines the information of human visual perception with that of the surface geometry, and gives a surface representation more informative to the human eye. The corresponding works have been published in [277, 278, 279].

Fourth, a novel saliency detection method has been developed: vertex-based saliency (VBS). The proposed vertex-based saliency measure is inspired by Lee's *mesh saliency* technique. This method differs from mesh saliency in making use of two other surface invariants to replace the multi-scale curvature analysis: the Retinex-based importance feature (RIF), and relative distance (RD), as we set out in Chapter 4. Together, these measures capture all the important aspects of the total information: the geometry of the surface. Thus, the proposed vertex-based method not only considers the geometry, but also takes into account the vagaries of human visual perception. The corresponding works have been published in [225, 256, 280].

Fifth, another novel saliency detection - region-based saliency (RBS) - was provided by extension from our work on VBS. The detection capability of VBS method is affected by occlusion and holes, and considered in isolation cannot guarantee completeness of the salient regions: in other words, they create fragmented regions. In estimating the saliency, they usually only consider local information: as a result, the yielded salient region may lose global information, which sometimes leads to unfavorable results, such as incomplete detection of salient regions. To this end, the RBS is proposed for superior stability and continuity. The strategy of the RBS is that it divides the original surface into regions, or patches, to which vertices are attached to improve saliency detection. This region-based method considers both local details, and the global shape information. Thus, it provides more reliable performance results. The corresponding works have been published in [225, 281].

Sixth, we demonstrated that surface saliency is useful for 3D surface applications that benefit from detection of the important regions of the surface. Surface simplification

was generated by focusing on these salient regions. Most of the existing simplification techniques are geometrically-based, which risks making further processing pointless, as the distinguishing characteristics of the object are rapidly lost. The goal of saliency-guided simplification is to delay the simplification of the salient parts of the surface, while strengthening simplification of the non-salient regions. In other words, this simplification approach preserves distinctive features better than other geometrically-based approaches: the distinctive features of each model should be simplified less than the rest of the mesh. The corresponding works have been published in [225, 256, 280].

Seventh, a saliency-guided interest points detection was implemented. The interest points were detected from the salient regions. Most existing methods focus on the geometric properties, such as curvature. However, the keypoints represent interesting information at fine scales and thus, may be sensitive to noise and other transformations. It was necessary to find larger interesting structures to overcome this problem. In our work, we detect the interest points using the detected salient regions. Our saliency detection methods are robust to change of viewpoint and noise corruption, and they have the ability to detect the same salient regions of the model under different data transformation instances. For this reason, according to our points detection criteria, the proposed points detector would detect the same interest points under data transformations. The corresponding works have been published in [278, 281].

8.3 Publication

Conference Papers

1. **[ICIAP2013]** Yitian Zhao, Yonghuai Liu, A 3D Interest Points Detection via Symmetric Surround based Surface Saliency. In Proceedings of the 17th International Conference on Image Analysis and Processing, (ICIAP), 2013. Lecture Notes in Computer Science Volume 8156, pp 632-641, 2013.

2. **[CAIP2013]** Yitian Zhao, Yonghuai Liu, Ziming Zeng: Using Region-based saliency for 3D interest points detection. In Proceedings of the 15th International Conference on Computer Analysis of Images and Patterns, (CAIP), Lecture Notes in Computer Science Volume 8048, pp 108-116, 2013.(Oral)

3. **[ICPR2012]** Yitian Zhao, Yonghuai Liu: Patch based saliency detection method for 3D surface simplification. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR), 2012, pp. 845-848.

4. [**ICIP2012**] Ran Song, Yonghuai Liu, Yitian Zhao, Ralph R. Martin, Paul L. Rosin: Conditional random field-based mesh saliency. In Proceedings of the International Conference on Image Processing (ICIP), 2012, pp. 637-640.(Oral)

5. [**ICIP2012**] Yitian Zhao, Yonghuai Liu, Ran Song, Min Zhang: Extended non-local means filter for surface saliency detection. In Proceedings of the International Conference on Image Processing (ICIP), 2012, pp. 633-636.(Oral)

6. [**ICASSP2012**] Yitian Zhao, Yonghuai Liu, Ran Song, Min Zhang: A saliency detection based method for 3D surface simplification. In Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2012, pp. 889-892.(Oral)

7. [**3DIMPVT2012**] Ran Song, Yonghuai Liu, Yitian Zhao, Ralph Martin, Paul L. Rosin: An Evaluation Method for Multiview Surface Reconstruction Algorithms. In Proceedings of the 3D Imaging Modeling Processing Visualization Transmission, (3DIMPVT), 2012, pp. 387-394.

8. [**ICASSP2011**] Ran Song, Yonghuai Liu, Yitian Zhao, Ralph R. Martin, Paul L. Rosin: MRF-based automatic image ordering and its application to mosaicing. In Proceedings of the 37th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2011, pp. 1549-1552.(Oral)

9. [**SoCPaR2011**] Yitian Zhao, Yonghuai Liu, Kezong Song, Min Zhang, Ran Song: A Retinex theory based point sampling method. In Proceedings of the 3rd International Conference on Soft Computing and Pattern Recognition, SoCPaR, 2011, pp. 330-335.

10. [**ISPA2011**] Yitian Zhao, Yonghuai Liu, Song Ran, and Min Zhang. A Retinex theory based points sampling method for mesh simplification. In Proceedings of the 7th International Symposium on Image and Signal Processing and Analysis, (ISPA), 2011, pp. 230-235. (Oral),(Best student paper award)

Journal Papers

1. [**IJCISIMA**] Yitian Zhao, Yonghuai Liu, Ran Song and Min Zhang. A Retinex theory based points sampling method. International Journal of Computer Information Systems and Industrial Management Applications, Vol. 5, pp 1-10, 2012.

2. [**IIJC**] Yitian Zhao, Yonghuai Liu. A MRF based points sampling method for efficient range image registration. IFRSA's International Journal of Computing, Vol2, issue 3, July 2012.

Bibliography

[1] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the SIGGRAPH*, pages 209–221, 1997.

[2] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. *IEEE Transacation on Pattern Analysis and Machine. Intelligence*, 34(10):1915–1926, October 2012.

[3] C. Lee, A. Varshney, and D. Jacobs. Mesh saliency. *ACM Transactions on Graphics*, 24(3):659–666, 2005.

[4] C. Cagniart. A patch-based mesh optimization algorithm for partitioned meshes. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* , pages 1339–1346, 2010.

[5] P. J. Neugebauer. Reconstruction of real-world objects via simultaneous registration and robust combination of multiple range images. *International Journal of Shape Modeling*, pages 71–90, 1997.

[6] M. Engineering. Construction engineering and inspection service. *State Project No.700-99-0393*, 2006.

[7] F. Prieto, T. Redarce, R. Lepage, and P. Boulanger. A non contact cad - based inspection system. In *Proceeding of the International Conference on Quality Control by Artifical Vision*, pages 133–138, 1999.

[8] G. Sansoni, M. Trebeschi, and F. Docchio. State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors*, 9(1):568–601, 2009.

[9] H. Unten and K. Ikeuchi. Virtual reality model of koumokuten generated from measurement. In *Proceedings of the 10th International Conference on Virtual Systems and Multimedia*, pages 209–216, 2004.

[10] S. F. El-Hakim, P. Boulanger, F. Blais, and J.-A. Beraldin. A system for indoor 3d mapping and virtual environments. In *Proceedings of SPIE*, pages 21–35, 1997.

[11] C. Frh and A. Zakhor. Constructing 3d city models by merging ground-based and airborne views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 562–569, 2003.

[12] M. Levoy, K. Pulli, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3d scanning of large statues. In *Pcoceeding of the SIGGRAPH*, pages 131–144, 2000.

[13] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. In *Proceeding of the SIGGRAPH*, pages 438–446, 2000.

[14] K. Berisso and T. Ollison. Coordinate measuring machine variations for selected probe head configurations. *Journal of Industrial technology*, 26(1):1–10, 2010.

[15] S. Motovalli, V. Suharitdamron, and A. Alrashdan. Design model generation for reverse engineering using multi-sensors. *IIE Transactions*, 30(4):357–366, 1998.

[16] Y. Li and P. Gu. Free-form surface inspection techniques state of the art review. *Computer Aided Design*, 36:1395–1417, 2004.

[17] S. K. Nayar, M. Watanabe, and Noguchi M. Real-time focus range sensor. *IEEE Transacation on Pattern Analysis and Machine Intelligence*, 18(12):1186–1198, 1996.

[18] B. Horn. Shape from shading: a method for obtaining the shape of a smooth opaque onject from one view. *PhD thesis, MIT, Cambridge, MA, USA*, 1970.

[19] R. White and D. A Forsyth. Combineing cues: shape from shading and texture. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1809–1816, 2006.

[20] L. H. Quam. Hierarchical warp stereo. In *Proceeding of the Image Understanding workshop*, pages 149–155, 1984.

[21] R. J. Woodlham. Analysing images of curved surface. *Artifical. Intelligence*, 17: 117–140, 1981.

[22] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal on Computer Vision*, 47 (1-3):7–42, 2002.

[23] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereorig? In *Proceedings of the European Conference on Computer Vision*, pages 563–578, 1992.

[24] H. C. Longuet-Higgins. A computer algorithm for re-constructing a scene from two projections. *Nature*, 293:133–135, 2009.

[25] F. Blais. A review of 2o years of range sensors development. *Journal of Electronic Imaging*, 13(1):231–240, 2004.

[26] VDI 2634. Part 2: Optical 3-d measuring systems optical system based on area scanning. 2002.

[27] VDI 2617. Part 6.2: Accuracy of coordinate measuring machines characteristics and their testing guideline for the application of din en iso 10360 to corrdinate measuring machines with optical distance sensors. *Beuth Verlag Gmbh*, 2005.

[28] B. Jahne, H.W. Haussecker, and P. Geissler. Sensors and imaging. *Handbook of Computer Vision and Applications*, 1999.

[29] M. C. Amann, T. Bosch, M. Lescure, and M. Rioux. Laser ranging: a critical review of usual techniques for distance measurement. *Journal of Optical Engineering*, 40(1):10–19, 2001.

[30] J. A. Beraldin, F. Blais, and U. Lohr. Laser scanning technology. *Chapter 1 of Airborne and Terrestrial Laser Scanning*, 2010.

[31] P. J. Besl. Active, optical range imaging sensors. *Optical Engineering*, 40(1):10–19, 2001.

[32] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. *Cambridge University Press*, 2000.

[33] T. M. Murali and T. A. Funkhouser. Consistent solid and boundary representations from arbitrary polygonal data. In *Proceeding of the Symposium on interactive 3D Graphics*, 1997.

[34] C. Shen, J. F. O'Brien, and J. R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *Proceedings of the SIGGRAPH*, pages 896–904, 2004.

[35] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of the SIGGRAPH*, pages 311–318, 1994.

[36] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:344–358, 1995.

[37] A. Hilton, A. J. Toddart, J. Illingworth, and T. Windeatt. Reliable surface reconstruction from multiple range images. In *Proceedings of the 4th European Conference on Computer Vision*, pages 117–126, 1996.

[38] E. Mcke. A robust implementation for three-dimensional delaunay triangulations. *Journal of Computational Geometry and Applications*, pages 70–73, 1995.

[39] J. Bloomenthal and B. Wyvill, editors. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. ISBN 155860233X.

[40] P. Shilane and T. Funkhouser. Selecting distinctive 3d shape descriptors for similarity retrieval. In *Proceeding of the Shape modelling international*, pages 18–27, 2006.

[41] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. *Computer & Graphics*, 26(2):65–70, July 1992.

[42] Kok-Lim Low and Tiow-Seng Tan. Model simplification using vertex-clustering. In *Proceeding of the 1997 symposium on Interactive 3D graphics*, pages 75–82, 1997.

[43] H. Hoppe. Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 99–108, 1996.

[44] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. *Computers & Graphics*, 22:37–54, 1997.

[45] D. P. Luebke. A developer's survey of polygonal simplification algorithms. *IEEE Computer graphics and applications*, 21:24–35, 2001.

[46] P. S. Heckbert and M. Garland. Survey of polygonal surface simplification algorithms, 1997.

[47] G. W. Qu, L. Meyer. Perceptually guided polygon reduction. *IEEE Transaction on Visualization and Computer Graphics*, 14:1015–1029, 2008.

[48] N. Williams, D. Luebke, J. D. Cohen, M. Kelley, and B. Schubert. Perceptually guided simplification of lit, textured meshes. In *Proceedings of the Symposium on Interactive 3D graphics*, pages 113–121, 2003.

[49] A. Bulbul, C. Koca, T. Capin, and U. Güdükbay. Saliency for animated meshes with material properties. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, pages 81–88, 2010.

[50] G. Leifman. Surface regions of interest for viewpoint selection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 414–421, 2012. ISBN 978-1-4673-1226-4.

[51] J. Theeuwes. Visual selective attention: a theoretical analysis. *Acta Psychologica*, 83:93–154, 1993.

[52] B. Arons. A review of the cocktail party effect. *Journal of the American voice I/O society*, 12(1):35–50, 2010.

[53] S. Frintrop, E. Rome, and H. I. Christensen. A review of the cocktail party effect. *ACM Transactions on Applied Perception*, 7:35–50, 1992.

[54] L. Bhatt, G. Groeger, K. McDermott, and T. Cotter. Rod and cone photoreceptor cells produce ros in response to stress in a live retinal explant system. *Molecular Vision*, 16:283–293, 2010.

[55] J. Herault and B. Durette. Modeling visual perception for image processing. In *Proceedings of the 9th international work conference on Artificial neural networks*, pages 662–675, 2007.

[56] N. Guyader, A. Chauvin, C. Massot, J. Herault, and C. Marendaz. *A biological model of low-level vision suitable for image analysis and cognitive visual perception.* Percetion, 2006.

[57] H. Teuber. Physiological psychology. *Annual Review of Psychology*, 6(1):247–248, 1955.

[58] A. M. Triesman and G. Gelade. A feature-integration theory of attention. cognitive psychology. *Annual Review of Psychology*, 12(1):97–136, 1980.

[59] R. Desimone and J. Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 19(6):247–248, 2009.

[60] S. K. Mannan, C. Kennard, , and M. Husain. The role of visual salience in directing eye movements in visual object agnosia. *Current biology*, 18(1):193–222, 1995.

[61] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis,. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20:1254–1259, 1998.

[62] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, 2009.

[63] D. Gao, S. Han, and N. Vasconcelos. Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *IEEE Transaction on Pattern Analysis Machince Intelligence*, 31:989–1005, 2009.

[64] F. Baluch and L. Itti. Mechanisms of top-down attention. *Trends in Neuroscience*, 34(4):210–224, 2011.

[65] L. Itti. Models of bottom-up attention and saliency. In *Tsotsos (Eds.), Neurobiology of Attention*, pages 576–582, 2005.

[66] D. Gao and N. Vasconcelos. Integrated learning of saliency, complex features, and object detectors from cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 282–287, 2005.

[67] M. Castelhano A. Torralba, A. Oliva and J. Henderson. Contextual guidance of eye movements and attention in realworld scenes: The role of global features on object search. *Psychological Review*, 113:766–786, 2006.

[68] W. Kienzle, F. A. Wichmann, M. O. Franz, and B. Scholkopf. A nonparametric approach to bottom-up visual saliency. In B. Scholkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 689–696. MIT Press, 2006.

[69] N. Bruce and J. Tsotsos. Saliency based on information maximization. *Advances in Neural Information Processing Systems*, 18:155–162, 2006.

[70] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10-12):1489–1506, 2000.

[71] E. Niebur and C. Koch. Computational architectures for attention. In R. Parasuraman, editor, *The Attentive Brain*, chapter 9, pages 163–186. MIT Press, Cambridge, MA, 1998.

[72] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *Proceeding of the International Conference on Computer Vision*, pages 2106–2113, 2009.

[73] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurbiology*, pages 219–227, 1985.

[74] J. J. Clark and N. J. Ferrier. Modal control of an attentive vision system. In *Proceedings of the IEEE International Conference on Computer Vision*, 1988.

[75] R. Milanese. Detecting salient regions in an image: from biological evidence to computer implementation. *PhD thesis*, 1993.

[76] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. *Advances in neural information processing systems*, 19:545–560, 2007.

[77] M. Cheng, G. Zhang, N. J. Mitra, X. Huang, and S. Hu. Global contrast based salient region detection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 409–416, 2011.

[78] M. Z. Aziz and B. Mertsching. Fast and robust generation of feature maps for region-based visual attention. *IEEE Transaction on Image Processing*, 17(5):633–644, May 2008.

[79] T. Avraham and M. Lindenbaum. Esaliency (extended saliency): Meaningful attention using stochastic image modeling. *IEEE Transacation on Pattern Analysis and Machine Intelligence*, 32(4):693–708, April 2010. ISSN 0162-8828.

[80] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[81] Z. Wang and B. Li. A two-stage approach to saliency detection in images. In *Prcoeedings of the IEEE International Conference on Acoustics Speech and Signal Processing* , pages 965–968, 2008.

[82] C. Guo and L. Zhang. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[83] X. Cui, Q. Liu, and D. Metaxas. Temporal spectral residual: fast motion saliency detection. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 617–620, 2009.

[84] H. J. Seo and P. Milanfar. Static and space-time visual saliency detection by self-resemblance. *Journal of Vision*, 9(12):1–27, 2009.

[85] N. D. B. Bruce and J. K. Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision*, 9(3):1–24, 2009.

[86] Li-Jia Li, Hao Su, Eric P. Xing, and Fei-Fei Li. Model simplification using vertex-clustering. In *Proceeding of the 1997 symposium on Interactive 3D graphics*, pages 75–82, 1997.

[87] D. Gao, V. Mahadevan, and N. Vasconcelos. On the plausibility of the discriminant center-surround hypothesis for visual saliency. *Journal of Vision*, 8(7), 2008.

[88] D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural Networks*, 19:1395–1407, 2006.

[89] L. Zhang, M.H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7), 2008.

[90] O. Le Meur, P. Le Callet, D. Barba, and D. Thoreau. A coherent computational approach to model bottom-up visual attention. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 28:802–817, 2006.

[91] J. Krauskopf, D. R. Williams, and D. W. Heeley. Cardinal directions of color space. *Vision Research*, 22:1123–1131, 1982.

[92] Y. F. Ma and H. J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *ACM Multimedia*, pages 374–384, 2003.

[93] Y. Zhai and M. Shah. Visual attention detection in video sequences using spatiotemporal cues. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 815–824, 2006.

[94] M. Miyahara and Y. Yoshida. Mathematical transform of (r,g,b) color data to munsell (h,s,v) color data. In *SPIE Proceedings : Visual Communications and Image Processing*, volume 1001, pages 650–657, San-Jose, 1988. SPIE.

[95] C. Guo and L. Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *Transaction on Image Processing*, 19:185–198, 2010.

[96] Y. Fang, W. Lin, B. Lee, C. Lau, Z. Chen, and C. Lin. Bottom-up saliency detection model based on human visual sensitivity and amplitude spectrum. *IEEE Transaction on Multimedia*, 14(1):187–198, February 2012. ISSN 1520-9210.

[97] S. Howlett, J. Hamill, and C. O'Sullivan. Predicting and evaluating saliency for simplified polygonal models. *ACM Transacation on Applied Perception*, 2(3):286–308, July 2005. ISSN 1544-3558.

[98] C. H. Tan. Saliency guided image-driven simplification. In *Proceeding of the 8th International Conference on Information, Communications and Signal Processing*, pages 1–4, 2011.

[99] M. Pauly and Gross M. Keiser, R. Multi-scale feature extraction on point-sampled surfaces. *Computer Graphics Forum*, 22:281–289, 2003.

[100] P. Longhurst, K. Debattista, and A. Chalmers. A gpu based saliency map for high-f andity selective rendering. In *Proceeding of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, pages 21–29, 2006.

[101] M. Feixas, M. Sbert, and F. Gonzalez. A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Transaction on Applied Perception*, 6(1):95–106, 2009.

[102] M. Novotni, P. Degener, and R. Klein. Correspondence generation and matching of 3d shape subparts. Technical Report CG-2005-2, June 2005.

[103] U. Castellani, M. Cristani, S. Fantoni, and V. Murino. Sparse points matching by combining 3d mesh saliency with statistical descriptors. *Computer Graphics Forum*, 27(2):643–652, 2008.

[104] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 373–380, 2009.

[105] R. Gal and Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics*, 25(1):130–150, 2006.

[106] Y. Kim and A. Varshney. Saliency-guided enhancement for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):925–932, 2006.

[107] R. Liu and H. Zhang. Segmentation of 3d meshes through spectral clustering. In *In The 12th Pacific Conference on Computer Graphics and Applications*, pages 3298–305, 2004.

[108] Y. Kim, A. Varshney, W. Jacobs, and François Guimbretière. Mesh saliency and human eye fixations. *ACM Transaction on Applied Perception*, 7(2):1–13, 2010. ISSN 1544-3558.

[109] H. Dutagaci, C. Cheung, and A. Godil. Evaluation of 3d interest point detection techniques via human-generated ground truth. *The Visual Computer*, 28:901–917, 2012.

[110] Y. Liu, M. Liu, D. Kihara, and K. Ramani. Salient critical points for meshes. In *Proceeding of the ACM symposium on Solid and physical modeling*, pages 277–282, 2007.

[111] R. Song, Y. Liu, Y. Zhao, R. Martin, and P. Rosin. Conditional random field-based saliency detection. In *Proceeding of the International Conference on Image Processing*, pages 637–640, 2012.

[112] R. Song, Y. Liu, R. R. Martin, and P. L. Rosin. 3d point of interest detection via spectral irregularity diffusion. *The Visual Computer*, 29(6-8):695–705, June 2013.

[113] J. Wu, X. Shen, W. Zhu, and L. Liu. Mesh saliency with global rarity. *Graphical Models*, 75:255–264, 2013.

[114] D. Parkhurst, K. Law, and E. Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision Research*, 42(1):107–123, 2002.

[115] J. Peng, V. Strela, and D. Zorin. A simple algorithm for surface denoising. In *Proceeding of the IEEE Visualization*, pages 107–112, 2001.

[116] M. Desbrun, M. Meyer, P. Schrroder, and A. H. Barr. Anisotropic feature-preserving denoising of height fields and bivariate data. In *Proceedings of Graphics Interface*, pages 145–152, 2000.

[117] B. Kim and J. Rossignac. Geofilter: Geometric selection of mesh filter parameters. *Computer Graphics Forum*, pages 295–302, 2005.

[118] C. Y. Chen and K. Y. Cheng. A sharpness dependent filter for mesh smoothing. *Computer Aided Geometric Design*, 22:376–391, 2005.

[119] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Transaction on Graphics*, 22(3):950–953, 2003.

[120] K. Hildebrandt and K. Polthier. Anisotropic filtering of non-linear surface features. In *Proceedings of the EuroGraphics*, pages 391–400, 2004.

[121] T. R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM Transaction on Graphics*, 22(3):943–949, July 2003. ISSN 0730-0301.

[122] G. Taubin. Geometric signal processing on polygonal meshes. In *Eurographics: State of the Art Report (STAR)*, page 0, 2000.

[123] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pages 351–358, 1995.

[124] A. Blake and A. Zisserman. Visual reconstruction. *Cambridge, Mass: MIT Press*, 1987.

[125] R. L. Stevenson and E. J. Delp. Viewpoint invariant recovery of visual surface from sparse data. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14 (9):257–270, 1992.

[126] J. Yi and D. Chelberg. Discontinuity-preserving and viewpoint invariant reconstruction of visible surface using a first order regularization. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17(6):624–629, 1995.

[127] L. Kobbelt. Discrete fairing. In *Proceedings of the Conference on the Mathematics of Surfaces*, page 101, 1997.

[128] L. Kobbelt. Discrete fairing and variational subdivision for freeform surface design. *The Visual Computing*, pages 142–158, 2000.

[129] Y. Sun, J. K. Paik, J. R. Price, and M. A. Abidi. Dense range image smoothing using adaptive regularization. In *Proceeding of the International Conference on Image Processing*, pages 744–747, 2000.

[130] Y. Sun, J. Paik, A. Koschan, and M. Abidi. Surface modeling using multi-view range and color images. *Integrated Computer Aided Engineering*, 10(1):37–50, January 2003.

[131] H. L. de Cougny, M. S. Shephard, and M. K. Georges. Explicit node point smoothing within the octree mesh generator. *SCOREC Report*, 10, 1990.

[132] S. A. Canann, M. B. Stephenson, and T. D. Blacker. Optismoothing: An optimization-driven approach to mesh smoothing. *Finite Elements in Analysis and Design*, 13(2-3):185–190, 1993.

[133] L. Freitag, M. Jones, and P. Plassmann. An efficient parallel algorithm for mesh smoothing. In *4th International Meshing Roundtable*, pages 47–58, 1995.

[134] A. Mezentsev. A generalized graph-theoretic mesh optimization model. In *Proceeding of the International Meshing Roundtable*, pages 255–264, 2004.

[135] N. Amenta, Bern. M, and D. Eppstein. Optimal point placement for mesh smoothing. In *Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms*, pages 528–537, 1997.

[136] J. Vollmer, R. Mencl, and H. Muller. Improved laplacian smoothing of noisy surface meshes. *Computer Graphics Forum*, 18(3):131–138, 1999.

[137] T. Zhou and K. Shimada. An angle-based approach to two-dimensional mesh smoothing. In *Proceedings of the 9th International Meshing Roundtable*, pages 373–384, 2000.

[138] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface smoothing via anisotropic diffusion of normals. In *Proceedings of the Conference on Visualization 2002*, pages 125–132, 2002.

[139] C. L. Bajaj and G. Xu. Anisotropic diffusion of surfaces and functions on surfaces. *ACM Transaction on Graphics*, 22(1):4–32, 2003.

[140] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[141] V. Parthasarathy and S. Kodiyalam. A constrained optimization approach to finite element mesh smoothing. *Finite Elements in Analysis and Design*, 9(1):302–320, 1991.

[142] A. Oddy, J. Goldak, M. McDill, and M. Bibby. A distortion metric for isoparametric finite elements. *Transactions of the Canadian Society for Mechanical Engineering*, 1988.

[143] A. Kalvin and Taylor. R. Superfaces: Polygonal mesh simplification with bounded error. *IEEE Computer Graphics and Applications*, 16(3):64–77, 1996.

[144] B. Chazelle, D. Dobkin, N. Shourhura, and A. Tal. Strategies for polyhedral surface decomposition: An experimental study. *Computational Geometry: Theory and Applications*, 7:327–342, 1997.

[145] V. Kraevoy, D. Julius, and A. Sheffer. Shuffler: Modeling with interchangeable parts. *Tech. Rep. TR-2006-09, Department of Computer Science, University of British Columbia*, 2006.

[146] G. Lavoue, G. Dupont, and A. Baskurt. New cad mesh segmentation method a, based on curvature tensor analysis. *Computer Aided Design*, 37(10):975–987, 2005.

[147] M. Eck, T. Derse, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of the SIGGRAPH*, pages 173–182, 1995.

[148] O. Sorkine, D. Cohen-Or, R. Goldenthal, and D. Lischinski. Bounded-distortion piecewise mesh parameterization. In *Proceeding of the IEEE Visualization*, pages 355–362, 2002.

[149] A. Mangan and R. Whitaker. Partitioning 3d surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 5(4): 308–321, 1999.

[150] D. Page, M. Abidi, A. Koschan, and T. Zhang. Object representation using the minima rule and superquadrics for under vehicle inspection. In *Proceeding of the 1st IEEE Latin American Conference on Robotics and Automation*, pages 91–97, 2003.

[151] Y. Sun, D. L. Page, J. K. Paik, A. Koschan, and M. A. Abidi. Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface

smoothing. In *Proceeding of the International Conference on Image Processing*, pages 825–828, 2002.

[152] J. Wu and L. Kobbelt. Structure recovery via hybrid variational surface approximation. *Computer Graphics Forum (Proceedings Eurographics 2005)*, 24(3):277–284, 2005.

[153] E. Zuckerberger, A. Tal, and S. Shlafman. Polyhedral surface decomposition with applications. *Computers & Graphics*, 26(5):733–743, 2002.

[154] L. Grady. Random walds for image segmentation. *IEEE Transaction on Pattern Analysis and Machine Interlligence*, 28(11):1768–1783, 2006.

[155] Y. Zhou and Z. Huang. Decomposing polygon meshes by means of critical points. In *Proceeding of the 10th International Multimedia ModellingConference*, 2004.

[156] K. Zhou, J. Syner, B Guo, and H. Shum. Isocharts: Stretch-driven mesh parameterization using spectral analysis. In *Proceedings of the 2004 Eurographics ACM SIGGRAPH symposium on Geometry processing*, pages 45–54, 2004.

[157] Y. Lai, S. Hu, R. Martin, and P. Rosin. Fast mesh segmentation using random walks. In *In Symposium on Solid and Physical modeling*, pages 183–191, 2008.

[158] Natasha Gelfand, Niloy J. Mitra, Leonidas J. Guibas, and Helmut Pottmann. Robust global registration. In *Proceedings of the third Eurographics symposium on Geometry processing*, 2005.

[159] A. Golovinskiy and T. Funkhouser. Randomized cuts for 3d mesh analysis. *ACM Transactions on Graphics*, 27(5):733–743, 2008.

[160] M. Mortara, G. Patane, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Blowing bubbles for multiscale analysis and decomposition of triangle meshes. *Algorithmica*, 38(1):227–248, 2003.

[161] M. Mortara, G. Patane, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Plumber: A method for a multi-scale decomposition of 3d shapes into tubular primitives and bodies. *ACM Symposium on Solid Modeling and Applications (2004),*, pages 2139–158, 2004.

[162] A. Sheffer. Model simplification for meshing using face clustering. *Computer Aided Design*, 33:925–934, 2001.

[163] R. O. Duda, P.E. Hart, and D. G. Stork. *Pattern Classification*. Wiley Interscience, 2nd edition, 2000.

[164] D. Cohen-Steiner, P. Alliez, and M Desbrun. Variational shape approximation. *ACM Transactions on Graphics*, 23(3):905–914, 2004.

[165] S. Shlafman, A. Tal, and S. Katz. Metamorphosis of polyhedral surfaces using decomposition. *Computer Graphics forum*, 21(3), 2002.

[166] D. Julius, V. Kraevoy, and A. Sheffer. D-charts: Quasi-developable mesh segmentation. *Computer Graphics Forum (Proceedings Eurographics 2005)*, 24(3): 981–990, 2005.

[167] M. Attene, S. Katz, M Mortara, and G. Patane. Mesh segmentationa comparative study. In *Proceedings of the Shape Modeling International*, pages 14–25, 2006.

[168] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. In *The Visual Computer (Proceedings of Pacific Graphics 2005)*, pages 865–875, 2005.

[169] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[170] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *ACMSIGGRAPH 2000*, pages 279–286, 2000.

[171] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249–259, March 2008.

[172] R. Liu and R. Zhang. Mesh segmentation via spectral embedding and contour analysis. In *Computer Graphics Forum*, 2007.

[173] A. Shamir. A formalization of boundary mesh segmentation. In *Proceeding of the second International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 82–89, 2004.

[174] B. Levy, S. Petitjean, and J. Ray, N. andMaillot. Least squares conformal maps for automatic texture atlas generation. In *Pcoceeding of SIGGRAPH*, pages 362–371, 2002.

[175] L. Moumoun, M. Chahhou, T. Gadi, and R. Benslimane. Normalized cuts and image segmentation. *International Journal of Engineering Science and Technology*, 1(2):3165–3171, 2010.

[176] Y. Lai, Q. Zhou, S. Hu, and R. Martin. Feature sensitive mesh segmentation. In *Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 17–25, 2006.

[177] H. Yamauchi, S. Lee, Y. Lee, Y. Ohtake, A. Belyaev, and H. Seidel. Feature sensitive mesh segmentation with mean shift. In *Proceedings of the International Conference on Shape Modeling and Applications 2005*, pages 238—245, 2005. ISBN 0-7695-2379-X.

[178] O. Sidi, O. van Kaick, Y. Kleiman, H. Zhang, and D. Cohen-Or. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Transaction on Graphics*, 30(6):126:1–126:10, December 2011.

[179] Q. Huang, V. Koltun, and L. Guibas. Joint shape segmentation with linear programming. *ACM Transaction on Graphics*, 30(6):125:1–125:12, December 2011.

[180] R. Ronfard and J. Rossignac. Full-range approximations of triangulated polyhedra. In *Proceeding of the Eurographics*, 1996.

[181] L. Kobbelt, S. Campagna, and H. Seidel. A general framework for mesh decimation. In *Proceedings of the Graphics Interface*, pages 43–50, 1998.

[182] R. Klein, A. Schilling, and W. Straber. Reconstruction and simplification of surfaces from contours. In *Proceedings of the 7th Pacific Conference on Computer Graphics and Applications*, pages 198–212, 1999.

[183] C. González, J. Gumbau, M. Chover, F. Ramos, and R. Quirós. User-assisted simplification method for triangle meshes preserving boundaries. *Computer Aided Design*, 41(12):1095–1106, December 2009.

[184] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 19–26, 1993.

[185] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, and W. Wright. Simplification envelopes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 119–128, 1996.

[186] P. Lindstrom. Out-of-core simplification of large polygonal models. In *Proceedings of SIGGRAPH*, pages 259–262, 2000.

[187] S. Zelinka and M Garland. Permission grids: Practical, error-bounded simplification. *Transactions on Graphcs*, 21(2), 2009.

[188] J. Talton. A short survey of mesh simplification algorithms. 24(3):277–284, 2005.

[189] J. Rossignac and P. Borrel. Multi-resolution 3d approximation for rendering complex scenes. In *Proceeding of the Geometric Modelling in Computer Graphics*, pages 453–465, 1993.

[190] P. Lindstrom. Out-of-core simplification of large polygonal models. In *Proceedings of the 27th Annual conference on Computer Graphics and Interactive Techniques*, pages 259–262, 2000. ISBN 1-58113-208-5.

[191] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 60(2):91–110, 2004. ISSN 0920-5691.

[192] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transaction on Graphics*, 30(1):1:1–1:20, February 2011.

[193] A. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2-3):348–361, September 2010.

[194] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *Prcoceedings of the IEEE 12th International Conference on Computer Vision Workshops*, pages 689–696, 2009.

[195] Y. Darom, T. Keller. Scale-invariant features for 3-d mesh models. *IEEE Transactions on Image Processing*, 21(5):2758–2769, 2012.

[196] J. Hu and J. Hua. Salient spectral geometric features for shape matching and retrieval. *Visual Computer*, 25:667–675, 2009.

[197] J. Novatnack and K. Nishino. Scale-dependent 3d geometric features. In *Proceeding of the IEEE 11th International Conference on Computer Vision,*, pages 1–8, 2007.

[198] G. Zou, J. Hua, Z. Lai, X. Gu, and M. Dong. Intrinsic geometric scale space by shape diffusion. *IEEE Transactions on Visualization and Computer Graphics,*, 15: 1193–1200, 2009.

[199] M. Schlattmann, R. Degener, and R. Klenin. Scale space based feature point detection on surfaces. *The Visual Computer*, 16:1–3, 2008.

[200] N. Walter, O. Aubreton, Y.D. Fougerolle, and O. Laligant. Susan 3D operator, principal saliency degrees and directions extraction and a brief study on the robustness to noise. In *Proceedings of the International Conference on Image Processing*, pages 3529–3532, 2009.

[201] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. V. Gool. Hough transform and 3d surf for robust three dimensional classification. In *European Conference on Computer Vision*, pages 589–602, 2010.

[202] F. Tombari, S. Salti, and L. Di Stefano. Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision*, 102(1-3):198–220, March 2013.

[203] H. Chen and B. Bhanu. 3d free-form object recognition in range images using local surface patches. *Pattern Recognition Letter*, 28(10):1252–1262, July 2007.

[204] Timor Kadir and Michael Brady. Saliency, scale and image description. *Int. J. Comput. Vision*, 45(2):83–105, 2001.

[205] L. Shao, T Kadir, and M Brady. Geometric and photometric invariant distinctive regions detection. *Information Sciences*, 177(4):1088–1122, 2007.

[206] C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens, and D. Vandermeulen. Feature detection on 3D face surfaces for pose normalisation and recogntiion. In *Proceeding of the IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–6, 2010.

[207] A. Godil and A.I. Wagan. Salient local 3D features for 3D shape retrieval. In *Proceedings of the 3D Image Processing (3DIP) and Applications II.*, pages 93–102, 2011.

[208] I. Sipiran and B. Bustos. Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visial Computer*, 27(11):963–976, November 2011.

[209] S. Salti, F. Tombari, and L. D. Stefano. A performance evaluation of 3d keypoint detectors. In *Proceeding of the 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 236–243, 2011. ISBN 978-0-7695-4369-7.

[210] X. Li and I. Guskov. Multi-scale features for approximate alignment. In *Proceedings of the Eurographics symposium on Geometry processing*, 2005.

[211] E. Land. Recent advances in retinex theory. *Vision Research*, 26(1):7–21, 1986.

[212] E. Land. The retinex. *American Scientist*, 52:283–293, 1964.

[213] Y. Liu, L. De Dominicis, R. R. Martin, and B. Wei. Illumination normalization using weighted gradient integral images. In *Proceeding of the International Conference on Image, Vision and Computing*, pages 329–333, 2013.

[214] Y. Park, S. Park, and J. Kim. Retinex method based on adaptive smoothing for illumination invariant face recognition. *Signal Processing*, 88(8):1929–1945, August 2008.

[215] M. Elad. Retinex by two bilateral filters. *Scale Space and PDE Methods in Computer Vision, Lecture Notes in Computer Science*, 3459:217–229, 2005.

[216] Jean Michel Morel, Ana Belén Petro, and Catalina Sbert. A pde formalization of retinex theory. *Trans. Img. Proc.*, 19(11):2825–2837, 2010.

[217] A. Hurlbert. Formal connections between lightness algorithms. *Journal of the Optical Society of America*, 3:1684–1692, 1986.

[218] D. Brainard and B. Wandell. Analysis of the retinex theory of color vision. *Vision Research*, 3:1651–1661, 1986.

[219] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel. A variational framework for retinex. *International Journal of Computer Vision*, 52:7–23, 2002.

[220] D. Jobson, Z. Rahman, and G. Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transaction on Image Processing*, 6(7):965–976, 1997.

[221] D. Jobson, Z. Rahman, and G. Woodell. Properties and performance of a center/surround retinex. *IEEE Transaction on Image Processing*, 6(3):451–462, 1997.

[222] Z. Rahman, D. Jobson, and G. Woodell. Retinex processing for automatic image enhancement. In *Proceeding of the IS&T SPIE Electronic Imaging 2002. The Human Vision and Electronic Imaging VII Conference*, pages 390–401, 2002.

[223] L. Meylan and S. Susstrunk. High dynamic rnage image rendering with a retinex-based adaptive filter. *IEEE Transaction on Image Processing*, 15(9):2820–2830, 2002.

[224] L. Meylan and S. Susstrunk. Bio-inspired image enhancement for natural color image. In *Proceeding of the IS&T SPIE Elctronic Imaging*, pages 46–56, 2004.

[225] Y. Zhao and Y. Liu. Patch based saliency detection method for 3d surface simplification. In *Proceeding of the International Conference on Pattern Recognition*, pages 845–848, 2012.

[226] L. Yang, X. Lu, W. Zeng, and W. Geng. Trilateral filtering-based retinex for image enhancement. In *Prcoeeding of the 4th international conference on Artificial Intelligence and Computational Intelligence*, pages 400–407, 2012.

[227] F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 177–187, 2003.

[228] J. Goldfeather and V. Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Transacation on Graphics*, 23(1):45–63, January 2004.

[229] M. Meyer, M. Desbrun, P. Schroder, and A. Barr. Discrete differential-geometry operators for triangulated 2-manifolds, 2002.

[230] J. J. Koenderink and Andrea J. van Doorn. Surface shape and curvature scales. *Image and Vision Computer*, 10(8):557–565, October 1992.

[231] H. Fan, Y. Yu, and Q. Peng. Robust feature-preserving mesh denoising based on consistent subneighborhoods. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):312–324, March 2010.

[232] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the International Conference on Computer Vision*, pages 839–845, 1998. ISBN 81-7319-221-9.

[233] A. Buades and and Morel J. Coll, B. A non-local algorithm for image denoising. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 60–65, 2005. ISBN 0-7695-2372-2.

[234] S. Yoshizawa, A. Belyaev, and H.r Seidel. Smoothing by example: Mesh denoising by averaging with similarity based weights. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications (2006)*, pages 38–44, 2006.

[235] G. Farin. Curves and surfaces for computer aided geometric design: A practical guide. *Fourth Ed. Academic Press, San Diego*, 10, 1997.

[236] A. Buades, B. Coll, and J. M. Morel. On image denoising methods. Technical report, Technical Note, CMLA (Centre de Mathematiques et de Leurs Applications, 2004.

[237] G. Mahmoudi, M.; Sapiro. Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Processing Letters*, 12:839–842, 2005.

[238] A. Lukin. A multiresolution approach for improving quality of image denoising algorithms. In *Proceeding of the International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[239] S. Kinderman, S. Osher, and P.W. Jones. Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling and Simulation*, 4(1):1091–1115, 2005.

[240] N. Azzabou, N. Paragios, and F. Guichard. Random walks, constrained multiple hypothesis and image enhancement. In *Proceedings of the European Conference on Computer Vision*, 2006.

[241] P. Couppe, P. Yger, and C. Barillot. Fast non-local means denoising for 3d mr images. In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 33–40, 2006.

[242] R. E. Barnhill and R. F. Riesenfeld. Computer aided geometric design. *Academic Press*, 1(1):1–60, 1984.

[243] S. Hahmann and S. Konz. Fairing bicubic b-spline surfaces using simulated annealing. In *Proceedings of the Curves and surfaces with applications in CAGD*, pages 159–168, 1997.

[244] P. Kiciak. Bicubic b-spline blending patches with optimized shape. *Computer Aided Design*, 43(2), February 2011.

[245] D. Rogers. An introduction to nurbs:with historical perspective. pages 179–203, 2001.

[246] H. Yagou, Y. Ohtake, and A. Belyaev. Mesh smoothing via mean and median filtering applied to face normals. In *Proceeding of the Geometric Modeling and Processing — Theory and Applications*, pages 124–130, 2002.

[247] Y. Shen and K. E. Barner. Fuzzy vector median-based surface smoothing. *IEEE Transaction on Visualization and Computer Graphics*, 10:252–265, 2004.

[248] A. Nehorai. Vector-sensor array processing for electromagnetic source localization. *IEEE Transactions on Signal Processing*, 42(2):376–398, 1994.

[249] Y. Hel-Or. 3d distance metric for pose estimation and object recognition for 2d projections. Technical report, Jerusalem, Israel, Israel, 1994.

[250] A. B. Yutaka and E. B. Y. Ohtake. A comparison of mesh smoothing methods. In *Proceedings of the Israel-Korea BiNational Conference on Geometric Modeling and Computer Graphics*, pages 83–87, 2003.

[251] S. J. Kim, W. K. Jeong, and C. H. Kim. Lod generation with discrete curvature error metric, 1999.

[252] N. Aspert, D. Santa-cruz, and T. Ebrahimi. Mesh: Measuring errors between surfaces using the hausdorff distance. In *Proceedings of the International Conference of Multimedia and Expo*, pages 705–708, 2002.

[253] P. Cignoni, C. Rocchini, and R. Scopogno. Metro: measuring error on simplified surfaces. *Computer Graphics forum*, 17(2):167–174, 1998.

[254] X. Sun, P. Rosin, R. Martin, and F. Langbein. Fast and effective feature-preserving mesh denoising. *IEEE Transactions on Visualization and Computer Graphics*, 13 (5):925–938, 2007.

[255] D. Hoffman and M. Singh. Salience of visual parts. *Cognition*, 69:29–78, 1997.

[256] Y. Zhao, Y. Liu, R. Song, and M. Zhang. A saliency detection based method for 3d surface simplification. In *Proceeding of the International Conference on Acoustics, Speech, and Signal Processing*, pages 889–892, 2012.

[257] L. Wolf, X. Huang, I. Martin, and D. Metaxas. Patch-based texture edges and segmentation. In *Proceedings of the European Conference on Computer Vision*, 2006.

[258] L. Liang, C. Liu, Y. Xu, B. Guo, and H. Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, 20(3):127–150, 2001.

[259] D. Heeger. Pyramid-based texture analysis/synthesis. In *Proceedings of the SIGGRAPH*, pages 229–238, 1995.

[260] A. Efros and W. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the SIGGRAPH*, pages 341–346, 2001.

[261] J. Portilla and E. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1): 49–70, 2009.

[262] I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-based image completion. In *Proceedings of the SIGGRAPH*, pages 303–312, 2003.

[263] S. P. Awate and R. T. Whitaker. Image denoising with unsupervised, information-theoretic, adaptive filtering. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 44–51, 2004.

[264] E. Ullman and E. Sali. Object classification using a fragment-based representation. *Lecture Notes in Computer Science*, 1811(3):73–87, 2000.

[265] W. Freeman, T. Jones, and E. Pasztor. Example-based super-resolution. *Computer Graphics and Applications*, 22(2):56–65, 2002.

[266] N. Voshell, L. Diachin, P. Knupp, and T. Munson. A patch-based mesh optimization algorithm for partitioned meshes. In *Proceedings of the 9th International Workshop on State-of-the-Art in Scientific and Parallel Computing*, 2008.

[267] H. Lin, W. Chen, and H. Bao. Adaptive patch-based mesh fitting for reverse engineering. *Computer-aided Design*, 39:1134–1142, 2007.

[268] T. Yamasaki and K. Aizawa. Patch-based compression for time-varying meshes. In *Proceedings of the 17th IEEE International Conference on Image Processing*, pages 3433–3436, 2010.

[269] Y. Zhang, C. Bajaj, and B. Sohn. Adaptive and quality 3d meshing from imaging data. In *Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 286–291, 2003.

[270] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transacation on Graphics*, 21(4):807–832, October 2002.

[271] A. Kaufman and E. Shimony. 3d scan-conversion algorithms for voxel-based graphics. In *Proceedings of the workshop on Interactive 3D graphics*, pages 45–75, 1987.

[272] W. Li and S. McMains. A gpu-based voxelization approach to 3d minkowski sum computation. In *Proceedings of the 14th ACM Symposium on Solid and Physical Modeling*, pages 31–40, 2010.

[273] N. Gagvani and D. Silver. Shape-based volumetric collision detection. In *Proceedings of the IEEE symposium on Volume visualization*, pages 57–61, 2000.

[274] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal on Comput. Vision*, 37(2):151–172, June 2000.

[275] Z. Lian, A. Godil, T. Fabry, T. Furuya, J. Hermans, R. Ohbuchi, C. Shu, D. Smeets, P. Suetens, D. Vandermeulen, and S. Wuhrer. Shrec'10 track: non-rigid 3d shape retrieval. In *Proceedings of the 3rd Eurographics conference on 3D Object Retrieval*, pages 101–108, 2010.

[276] H. Dutagaci, A. Godil, P. Daras, A. Axenopoulos, G. Litos, S. Manolopoulou, K. Goto, T. Yanagimachi, Y. Kurita, S. Kawamura, T. Furuya, and R. Ohbuchi. Shrec'11 track: generic shape retrieval. In *Proceedings of the 4th Eurographics conference on 3D Object Retrieval*, pages 65–69, 2011.

[277] Y. Zhao, Y. Liu, R. Song, and M. Zhang. A retinex theory based points sampling method for mesh simplification. In *Proceeding of the International Symposium on Image and Signal Processing and Analysis*, pages 230–235, 2011.

[278] Y. Zhao and Y. Liu. A 3d interest points detection via symmetric surround based surface saliency. In *Proceeding of the International Conference on Image Analysis and Processing*, pages 632–641, 2013.

[279] Y. Zhao, Y. Liu, R. Song, and M. Zhang. A retinex theory based points sampling method. *International Journal of Computer Information Systems and Industrial Management Applications*, 5:1–10, 2012.

[280] Y. Zhao, Y. Liu, R. Song, and M. Zhang. Extended non-local means filter for surface saliency detection. In *Proceeding of the International Conference on Image Processing*, pages 633–636, 2012.

[281] Y. Zhao, Y. Liu, and Z. Zeng. Using region-based saliency for 3d interest points detection. In *Proceeding of the International Conference on Image Analysis and Processing*, pages 108–116, 2013.